



Universidad
Carlos III de Madrid

Departamento de Informática
Ing. Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

Análisis de requisitos en el desarrollo del software

Autor: Dalila García Notario

Tutor: Miguel Ángel Ramos González

Leganés, Octubre de 2015

ESTA HOJA SE HA DEJADO EN BLANCO INTENCIONADAMENTE

Título: Análisis de requisitos en el desarrollo del software

Autor: Dalila García Notario

Director: Miguel Ángel Ramos González

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 05 de Octubre de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Llegados a este punto, una vez finalizado el proyecto, llega la hora de pararme a pensar por una parte en todo el esfuerzo que me ha costado realizar este proyecto ya que para mí ha sido súper duro el compaginar el proyecto con los duros turnos de mi trabajo, donde ha habido noches de desesperación por no ver avance, madrugones tremendos sin importar que fuera festivo, sábado o lunes y por otra parte me paro en pensar en toda la gente que me ha ayudado de manera directa o indirectamente para llegar hasta aquí.

A mí, personalmente se me pasan muchísimas personas por la cabeza, de las cuales evidentemente no puedo mencionar a todos, pero si haré una mención especial a los más importantes para mí. A la hora de mencionarlos no daré ningún tipo de prioridad, ya que considero que todos han aportado un “granito de arena” para alcanzar el objetivo.

Me gustaría hacer una mención especial a mi tutor, Miguel Ángel Ramos por su eterna paciencia, apoyo incondicional, por su rapidez y eficiencia sin importarle que fuera sábado o domingo ni que fueran las siete de la mañana ni las doce de la noche.

Seguidamente, me gustaría agradecer a mis padres Begoña y Javi por haber luchado cada día para que fuera mejor persona gracias a sus consejos. Gracias por su apoyo económico, y más importante, su apoyo moral, ya que sin estos no hubiera podido ni hubiera tenido el valor de llegar hasta aquí. Pensándolo fríamente, los dos han formado un buen equipo.

A Yael, mi hermana, que a pesar de ser la pequeña y revoltosa de la casa, me ha ayudado mucho a avanzar solamente haciéndome de rabiar, consiguiendo que dejara de pensar en ello.

Por otra parte a mi tío Antonio, que más que mi tío parece mi hermano, por ser una de las mejores personas que conozco, por ser mi guía y mi ejemplo a seguir y por ser uno de mis confesores favoritos 😊.

No podría dejar de mencionar a mis abuelos maternos, Carmen y Antonio, por ser unas de las personas más especiales para mí y por haber ejercido en muchos momentos de madre y padre y particularmente a mi abuelo, a quien dedico este proyecto por estar pasando por malos momentos de salud, además de ser probablemente la persona que más ánimos y buenos consejos me ha dado a lo largo de mi vida, sin dudar ni un momento de mí.

A Iván, mi chico, por ser sobre todo mi amigo y al que nunca le podre podré agradecer todo lo que hizo por mí. Millones de gracias por tu cariño, comprensión y apoyo incondicional. ¡Cada abrazo, cada mirada, cada beso, cada sonrisa...make me in love! ¡Te quiero mil!

Y finalmente a Arantxa, mi amiga, la que probablemente más ha sufrido mis agobios, llantos y risas a lo largo de este proyecto y de todos los años que llevamos compartiendo. ¡Que dure la amistad!

Con esto, solo decir que ¡millones de gracias a todos!

ESTA HOJA SE HA DEJADO EN BLANCO INTENCIONADAMENTE

Resumen

El tema principal de este proyecto, como su nombre indica, son los requisitos software. Como se puede observar por el número de páginas de la memoria es un tema bastante amplio aunque a priori por su nombre parece ser un tema que carece de importancia, idea muy equivocada.

El proyecto está formado por una serie de capítulos, en concreto dieciocho, donde cada uno de ellos se centra en una tarea o actividad relacionada con la gestión de requisitos software dentro de un proyecto.

El proyecto estudia todos los aspectos relacionados con los requisitos software, la evolución desde que estos son obtenidos, englobando las diferentes técnicas existentes, las clasificaciones de los diferentes tipos de requisitos software que existen hasta el punto en el que estos son validados y plasmados en el documento de especificación de requisitos software.

Además, a lo largo del proyecto se ha puesto especial énfasis en la gran importancia de realizar una correcta especificación de requisitos software, así como emplear el tiempo necesario aportando las pautas oportunas a seguir para alcanzar el objetivo, incluyendo en estas buenas prácticas como por ejemplo la inclusión de un resumen, la realización de un caso de negocio o la redacción del documento de especificación de requisitos fácil de modificar por destacar alguna.

Debido a que la especificación de requisitos está unida con la calidad del producto, también podemos encontrar un capítulo dedicado a esto, donde se tocan aspectos como las técnicas más comunes para verificar la calidad o se realiza una mención especial a los principales y secundarios estándares relacionados con ello.

Aparte de todo esto anterior y debido a la gran importancia que tienen actualmente en el mercado, se relaciona la gestión de requisitos con guías y modelos de gran importancia como son PMBOK y CMMI.

Y como punto final a esto, se han realizado dos casos prácticos, uno de ellos relacionado con calidad, y que consiste en una encuesta de satisfacción a los agentes para verificar la calidad del sistema y de la compañía en general, esté relacionado con el sector eléctrico en el que trabajodesdehace tiempo, y otro caso práctico que consiste en una posible mejora desde mi punto de vista del sistemaactual de casación del mercado eléctrico diario, y donde se han definido algunos de los requisitos más importantes.

Resumiendo se ha puesto especial atención tanto en los requisitos software, que es el tema principal del proyecto, como temas derivados de este que he considerado de importancia.

Palabras clave: requisitos software, calidad, CMMI, IEEE, proyectos software, documento de especificación de requisitos software, validación requisitos.

ESTA HOJA SE HA DEJADO EN BLANCO INTENCIONADAMENTE

Abstract

The main subject of this Project, as the name already implies, are Software Requirements. Firstly it can be thought as a not important subject, but that is a wrong idea, this is why the extension of the report is wide.

The project is building with a serial of chapters, exactly eighteen, where each of all is focused on a specific task or activity related to management of software requirements in a project.

The project studies all the aspects which involves software requirements, the evolution from the beginning, containing all existing techniques, the classification of the different kind of software requirements until them are validated and added to the SRS document (Software Requirement Specification).

Furthermore, special emphasis has been placed on preparing a correctly software requirement specification (SRS), as well as the use of the required time to reach the target, including good practices such as including a summary, preparing a business case or write an easy to modify requirements specification document , to highlight some of them.

Due to the requirement specification is linked to the final quality of the product, also it can find a chapter related to this, where more common techniques for verifying the quality are showed, but also it is mentioned other standards.

Because of the huge important of the market it's necessary connect the management of the requirements with guides and models like are PMBOK and CMMI.

Finally it has been realized two case of study, one related with quality: a satisfaction survey for the agents trying to verify the system quality and the company in general of the electricity market where I actually work. And another case of study that consists in introducing a possible update in the current system of the matching of daily electricity market, and where I used most of the important requirements.

Summarizing it has given special attention in software requirements which is the main subject of the project, as in issues arising from it that I have considered them relevant.

Keywords: software requirements, quality, CMMI, IEEE, software project, software requirement specification document, validation of requirements

Índice general

Contenido

Capítulo 1: Introducción y objetivos	16
1.1 Introducción	16
1.2 Objetivos	17
1.3 Fases de desarrollo.....	18
1.4 Medios empleados	18
1.5 Estructura de la memoria.....	19
Capítulo 2: Proyectos software	21
2.1 Gestión de proyectos software.....	23
2.2 Modelos del proceso del software.....	25
Capítulo 3: Ingeniería de requisitos	30
3.1 Clasificación según Ian Sommerville	31
3.1.1 Estudios de viabilidad.....	32
3.1.2 Obtención y análisis de requisitos.....	33
3.1.3 Especificación de requisitos	34
3.1.4 Validación de requisitos	38
3.2 Clasificación según Roger Pressman	39
3.2.1 Concepción	39
3.2.2 Indagación	39
3.2.3 Elaboración.....	40
3.2.4 Negociación	40
3.2.5 Especificación	40
3.2.6 Validación	40
3.2.7 Administración de los requisitos	41
Capítulo 4: Requisitos.....	42
4.1 Definición de requisitos.....	42
4.2 Actividades que tienen relación con los requisitos.....	42
4.3 Información acerca de las necesidades del cliente	43
4.4 Documento especificación de requisitos software	43
4.5 Gestión de requisitos.....	44
4.5.1 Prácticas y principios relacionados con la gestión de requisitos	44

4.5.2	Procedimientos asociados a la gestión de requisitos.....	45
4.6	Control de versiones en el conjunto de requisitos.....	46
Capítulo 5: Documento especificación de requisitos software.....		47
5.1	¿Qué es? ¿Por quién es utilizado?.....	47
5.2	Documento de especificación de requisitos software con estructura estándar	48
5.2.1	Beneficios de utilizar estructura estándar.....	48
5.2.2	Mejor manera de implementar el documento con estructura estándar	48
5.2.3	Costos y problemas asociados al uso del documento con estructura estándar	50
5.3	Pautas de cómo utilizar el documento de especificación de requisitos	50
5.4	Inclusión de un resumen de los requisitos.....	51
5.4.1	Beneficios que aporta el resumen.....	51
5.4.2	Costos y problemas asociados al uso del resumen	52
5.5	Realización de un caso de negocio para el sistema.....	52
5.6	Definición de un glosario de términos	53
5.6.1	Definición de glosario de términos	53
5.6.2	Beneficios que aporta el glosario a los usuarios del documento.....	53
5.6.3	Costos y problemas asociados al uso del glosario.....	54
5.7	Diseño del documento para facilitar la lectura	54
5.8	Creación de un documento de especificación de requisitos fácil de cambiar	55
Capítulo 6: Contenido del documento de especificación de requisitos.....		56
6.1	Introducción	57
6.1.1	Propósito	57
6.1.2	Alcance	58
6.1.3	Definiciones, acrónimos y abreviaturas	59
6.1.4	Referencias.....	59
6.1.5	Visión global	60
6.2	Descripción general.....	60
6.2.1	Perspectiva del producto.....	60
6.2.2	Funciones del producto.....	62
6.2.3	Características del usuario	62
6.2.4	Restricciones.....	63
6.2.5	Supuestos y dependencias	64
6.2.6	Distribución de requisitos.....	64
6.3	Requisitos específicos.....	64

6.3.1	Requisitos funcionales.....	65
6.3.2	Requisitos de interfaz externa.....	66
6.3.3	Requisitos de ejecución.....	66
6.3.4	Requisitos lógicos de la base de datos	67
6.3.5	Restricciones de diseño	67
6.3.6	Información de apoyo	76
Capítulo 7: Obtención de requisitos.....		77
7.1	Problemas asociados a la obtención e interpretación de los requisitos.....	77
7.2	Modelo general del proceso de obtención y análisis de requisitos	77
7.3	Técnicas para realizar la obtención de requisitos	79
7.4	Descripción de requisitos	84
Capítulo 8: Tipos de requisitos		88
8.1	Requisitos funcionales.....	88
8.2	Requisitos no funcionales.....	88
8.3	Requisitos del usuario	90
8.4	Requisitos del sistema	91
8.5	Requisitos del dominio	92
Capítulo 9: Modelo CMMI		93
9.1	Introducción al modelo CMMI	93
9.2	Desarrollo de requisitos del producto y requisitos de componente de producto.....	94
9.2.1	Desarrollo de requisitos del producto y requisitos de componente de producto.....	96
9.3	Gestión de requisitos con CMMI.....	96
9.3.1	Interpretación de requisitos.....	97
9.3.2	Obtención del compromiso sobre los requisitos.....	98
9.3.3	Tramitar modificaciones a los requisitos.....	98
9.3.4	Sustentar la trazabilidad bidireccional de los requisitos.....	98
9.3.5	Garantizar el alineamiento entre los requisitos del trabajo y el proyecto.....	99
9.4	Análisis y validación de requisitos con CMMI	99
9.4.1	Determinación de escenarios y conceptos de operación	99
9.4.2	Determinación de la definición de funcionalidad y de las particularidades o atributos de calidad necesarios	100
9.4.3	Análisis de requisitos.....	100
9.4.4	Análisis de requisitos con el fin de alcanzar un equilibrio.....	100
9.4.5	Validar los requisitos	101

Capítulo 10: Sistemas críticos.....	102
10.1 Introducción	102
10.2 Especificación dirigida por riesgos	103
10.2.1 Análisis de riesgos	103
Capítulo 11: Validación de requisitos.....	107
11.1 ¿En qué consiste la fase de validación de requisitos?.....	107
11.2 Técnicas relacionadas con la validación de requisitos	107
11.2.1 Técnica basada en la elaboración de prototipos.....	107
11.2.2 Técnica basada en la verificación de requisitos	109
11.2.3 Técnica basada en la creación de casos de pruebas	111
11.2.4 Animación y realización de modelos	111
11.2.5 Realización de manuales de usuarios.....	111
11.3 Observaciones de interés en la fase de validación de requisitos.....	111
Capítulo 12: Calidad de proyectos software	115
12.1 Información general sobre calidad.....	115
12.1.1 Definición de calidad	115
12.1.2 Tipos de orígenes de la calidad	115
12.1.3 Tipos de vistas relacionadas con calidad.....	116
12.1.4 Actividades que engloba la gestión de calidad	117
12.2 Herramientas y técnicas de calidad.....	120
12.2.1 Herramientas básicas de calidad	121
12.2.2 Herramientas de gestión	127
12.2.3 Herramientas de creatividad.....	129
12.2.4 Herramientas estadísticas	130
12.2.5 Herramientas de diseño	131
12.2.6 Herramientas de medición.....	132
12.3 Normas y estándares de calidad	134
12.3.1 Normas relacionadas con calidad.....	134
12.3.2 Principios relacionados con las normas ISO9000	136
12.4 Calidad de productos software	137
12.4.1 Aspectos a destacar de la calidad de un producto software	138
12.5 Caso práctico: Encuesta de calidad	143
12.5.1 Introducción caso práctico	143
12.5.2 Contenido de la encuesta de calidad	143

15.5.3	Estudio de resultados obtenidos	150
Capítulo 13:	presupuesto	153
13.1	Introducción	153
13.1.1	Planificación	153
13.1.2	Planificación del proyecto mediante diagrama de Gantt.....	154
13.1.3	Presupuesto.....	156
Capítulo 14:	Glosario	158
Capítulo 15:	Siglas	166
Capítulo 16:	Bibliografía	167
Capítulo 17:	Conclusiones	169
17.1	Vías de investigación futuras.....	169
17.2	Opiniones personales	169
Capítulo 18:	Anexo	170
18.1	Introducción caso práctico	170
18.2	Sistemas necesarios.....	172
18.3	Análisis de los diferentes sistemas	173
18.3.1	Actores del sistema	173
18.3.2	Requisitos funcionales.....	174
18.3.3	Requisitos no funcionales.....	177

Índice de figuras

ILUSTRACIÓN 1: ESTIMACIÓN DE PROYECTOS EXITOSOS, FRACASADOS Y DEFICIENTES [15 DE SEPTIEMBRE DE 2014]	16
ILUSTRACIÓN 2: ESQUEMA DEL CICLO DE VIDA EN CASCADA	27
ILUSTRACIÓN 3: ESQUEMA DEL CICLO DE VIDA EN "V"	28
ILUSTRACIÓN 4: ESQUEMA DEL CICLO DE VIDA "ENTREGA INCREMENTAL"	29
ILUSTRACIÓN 5: ACTIVIDADES DE LA INGENIERÍA DE REQUISITOS	32
ILUSTRACIÓN 6: ACTIVIDADES QUE FORMAN LA GESTIÓN DE REQUISITOS.....	45
ILUSTRACIÓN 7: ESTRUCTURA DEL DOCUMENTO ERS SEGÚN IEEE-STD-830-1998.....	56
ILUSTRACIÓN 8: OPCIÓN 1, DOCUMENTO ERS CUANDO LA ORGANIZACIÓN DEPENDE DEL MODO DE OPERACIÓN	69
ILUSTRACIÓN 9: OPCIÓN 2, DOCUMENTO ERS CUANDO LA ORGANIZACIÓN DEPENDE DEL MODO DE OPERACIÓN	70
ILUSTRACIÓN 10: ESTRUCTURA DEL DOCUMENTO ERS CUANDO LA ORGANIZACIÓN DEPENDE DE LA CLASE DE USUARIO.....	71
ILUSTRACIÓN 11: DOCUMENTO ERS CUANDO LA ORGANIZACIÓN DEPENDE DEL OBJETO	72
ILUSTRACIÓN 12: EJEMPLO DE DIAGRAMA CUANDO LAS LIMITACIONES NO SE CUMPLEN	73
ILUSTRACIÓN 13: EJEMPLO DE DIAGRAMA QUE RELACIONA EL CERTIFICADO CON LAS CREDENCIALES UTILIZADAS.....	73
ILUSTRACIÓN 14: DOCUMENTO ERS CUANDO LA ORGANIZACIÓN DEPENDE DEL ESTÍMULO	74
ILUSTRACIÓN 15: DOCUMENTO ERS CUANDO LA ORGANIZACIÓN DEPENDE DE LA JERARQUÍA FUNCIONAL	75
ILUSTRACIÓN 16: PROCESO DE OBTENCIÓN Y ANÁLISIS DE REQUISITOS	78
ILUSTRACIÓN 17: PROCESO DE REALIZACIÓN DE PROTOTIPOS	82
ILUSTRACIÓN 18: CLASIFICACIÓN REQUISITOS NO FUNCIONALES	89
ILUSTRACIÓN 19: PROCESO DE VALIDACIÓN DE REQUISITOS	114
ILUSTRACIÓN 20: EJEMPLO DIAGRAMA DE FLUJO SOBRE LOS PROBLEMAS DE ACCESO AL WEB DE AGENTES	123
ILUSTRACIÓN 21: DIAGRAMA DE CONTROL	125
ILUSTRACIÓN 22: MODELOS UTILIZADOS PARA VALORAR LA CALIDAD DE PRODUCTOS SOFTWARE.....	138
ILUSTRACIÓN 23: TIPOS DE CALIDAD	138
ILUSTRACIÓN 24: GRÁFICA CORRESPONDIENTE A LOS INDICADORES OBJETIVOS.....	151
ILUSTRACIÓN 25: DIAGRAMA DE GANTT	155

Índice de tablas

TABLA 1: MODELO DE CICLO DE VIDA CODE-AND-FIX	26
TABLA 2: MODELO DE CICLO DE VIDA EN CASCADA	27
TABLA 3: MODELO DE CICLO DE VIDA EN "V"	27
TABLA 4: MODELO DE CICLO DE VIDA EN ESPIRAL	28
TABLA 5: MODELO DE CICLO DE VIDA "PROTOTIPADO"	28
TABLA 6: MODELO DE CICLO DE VIDA "ENTREGA INCREMENTAL"	29
TABLA 7: CUADRO COMPARATIVO SEGÚN LOS AUTORES IAN SOMERVILLE Y ROGER PRESSMAN.....	31
TABLA 8: EJEMPLO DE MODELO DE REFERENCIA	59
TABLA 9: INFORMACIÓN CONTENIDA EN UN PRODUCTO SOFTWARE	61
TABLA 10: CARACTERÍSTICAS GENERALES DE LOS USUARIOS	63
TABLA 11: EJEMPLO DE REQUISITOS SOFTWARE.....	65
TABLA 12: NOTACIONES A PODER UTILIZAR EN LA ERS.....	92
TABLA 13: TIPOS DE RIESGOS.....	104
TABLA 14: RELACIÓN ENTRE LOS FACTORES DISPONIBLES Y EL PROTOTIPO A UTILIZAR	108
TABLA 15: DETALLE DE ANOMALÍA Y POSIBLES ACCIONES A TOMAR PARA SOLVENTARLA	110
TABLA 16: DEFINICIONES DE LOS DIFERENTES TIPOS DE CALIDAD	116
TABLA 17: DEFINICIÓN DE LOS TIPOS DE VISTAS	117
TABLA 18: ACTIVIDADES QUE FORMAN LA GESTIÓN DE CALIDAD	117
TABLA 19: TIPOS DE REVISIONES	120
TABLA 20: HERRAMIENTAS Y TÉCNICAS DE CALIDAD	121
TABLA 21: SÍMBOLOS UTILIZADOS EN DIAGRAMAS DE FLUJOS.....	122
TABLA 22: FASES DE DESARROLLO DE UN "BENCHMARKING"	134
TABLA 23: CONJUNTO DE NORMAS QUE FORMAN LA ISO 9000.....	135
TABLA 24: NORMAS RELACIONADAS CON CALIDAD Y CON LAS NORMAS ISO 9000	136
TABLA 25: LA FUNCIONALIDAD COMO FACTOR DE CALIDAD EXTERNA E INTERNA	139
TABLA 26: LA FIABILIDAD COMO FACTOR DE CALIDAD EXTERNA E INTERNA	140
TABLA 27: LA USABILIDAD COMO FACTOR DE CALIDAD EXTERNA E INTERNA.....	140
TABLA 28: LA EFICIENCIA COMO FACTOR DE CALIDAD EXTERNA E INTERNA.....	140
TABLA 29: LA MANTENIBILIDAD COMO FACTOR DE CALIDAD EXTERNA E INTERNA.....	141
TABLA 30: LA PORTABILIDAD COMO FACTOR DE CALIDAD EXTERNA E INTERNA	142
TABLA 31: LA EFECTIVIDAD COMO FACTOR DE CALIDAD EN USO	142
TABLA 32: LA PRODUCTIVIDAD COMO FACTOR DE CALIDAD EN USO	142
TABLA 33: LA SEGURIDAD EN USO COMO FACTOR DE CALIDAD EN USO.....	142
TABLA 34: LA SATISFACCIÓN COMO FACTOR DE CALIDAD EN USO.....	142
TABLA 35: RESUMEN ESTADÍSTICO DE LAS RESPUESTAS DE LOS AGENTES.....	151
TABLA 36: PORCENTAJE CORRESPONDIENTE A LOS INDICADORES OBJETIVOS.....	152
TABLA 37: RELACIÓN ENTRE PORCENTAJES Y ACCIONES CORRECTIVAS A TOMAR.....	152

Capítulo 1: Introducción y objetivos

1.1 Introducción

En general, por los datos existentes desde hace tiempo, se sabe que el tiempo empleado en la fase de requisitos es mínimo para lo que se debería emplear, ya que según estudios ya realizados, más de la mitad de los errores detectados en un producto o sistema son generados en esta fase. Como consecuencia de esto, se produce un sobre coste en el proyecto, ya que el coste que implica arreglar un defecto una vez realizada la entrega del producto es extremadamente alto si lo comparamos con lo que hubiera costado si este hubiera sido detectado en la fase correspondiente.

En términos generales, esto va mejorando, ya que según el resultado de informes ofrecidos por Standish Group relativos al resultado de los proyectos “The Curious Case of the CHAOS Report 2009” [15 de Septiembre de 2014], a medida que pasan los años, dichos resultados son mejores respecto a años anteriores.

	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16%	27%	26%	28%	34%	29%	35%	32%
Challenged	53%	33%	46%	49%	51%	53%	46%	44%
Failed	31%	40%	28%	23%	15%	18%	19%	24%

Ilustración 1: Estimación de proyectos exitosos, fracasados y deficientes [15 de Septiembre de 2014]

Como se puede observar en la tabla anterior, esta contiene la información relativa al porcentaje de proyectos realizados con éxito, proyectos fracasados y proyectos deficientes para un intervalo de años que comprende desde el 1994 hasta el 2009.

Analizando el conjunto de información, se observa que los resultados obtenidos en el año 2009, comparados con los resultados del año 1994 son notablemente mejores. Se puede visualizar que en el año 2009 se tiene una tasa de éxito del 32% frente al 16% existente en 1994, así como también se ha de tener en cuenta que desde el año 1994 hasta el año que se poseen datos, la tasa de éxito es creciente en cada año. Por otro lado en la tasa de proyectos fracasados y proyectos deficientes, el porcentaje en el año 2009 es menor respecto al año 1994.

Aunque como se comenta, los resultados obtenidos cada año son mejores, aún queda mucho camino por recorrer y mucho que mejorar en este aspecto, ya que las cifras aún son muy bajas.

Parece obvio el definir el significado de proyecto realizado con éxito, proyecto fracasado o proyecto deficiente, ya que son definiciones que están dentro de nuestro alcance, no obstante debido a que estas tres terminologías se pueden enfocar de diferente manera, se definen a continuación:

- Proyecto realizado con éxito: según la definición aportada por Norberto Figuerola, un proyecto se considera exitoso cuando este se realiza en plazo y presupuesto y se cumplen los requisitos especificados en el alcance.

Además, según establece la guía de los fundamentos de la dirección de proyectos PMBOK (*Project Management Body of Knowledge*), para poder considerar que un proyecto ha sido exitoso se deben cumplir las siguientes expectativas:

- Nivel I. Alcanzar los objetivos del proyecto
 - Nivel II. Eficiencia del proyecto.
 - Nivel de interrupción del trabajo del cliente.
 - Eficiencia en el uso de los recursos
 - Crecimiento del número de miembros del equipo
 - Gestión de conflictos
 - Nivel III. Utilidad para el usuario/cliente final.
 - ¿Ha sido solucionado el problema inicial?
 - ¿Se han incrementado los beneficios o se ha producido ahorro real?
 - ¿El usuario se encuentra actualmente usando el producto?
 - Nivel IV. Mejora organizacional: Aprender sobre la experiencia
- Proyecto fracasado: proyecto que no ha llegado a terminarse, y por ello ha sido abortado antes de su finalización. Según informa la guía PMBOK, el grado de madurez de la organización y los procedimientos internos establecidos pueden contribuir al éxito o fracaso del proyecto. Es decir:
- ❖ Si la organización trabaja habitualmente en proyectos, se dispone de pautas ya definidas.
 - ❖ Vías de comunicación formal: Si son muy rígidas pueden entorpecer el trabajo
 - ❖ Vías de comunicación informal (amistades, conocidos, etc.): si son muy frecuentes pueden producir desinformación
- Proyectos deficientes: son aquellos proyectos en los que por alguna razón justificada debe modificarse alguno de sus parámetros, ya sea el alcance inicial, tiempo estimado, y/o coste planificado.

1.2 Objetivos

Como se ha explicado anteriormente, una no correcta definición de requisitos software implica cantidad de problemas, hablando desde problemas económicos (el coste de rectificar o solventar un problema es más elevado en etapas avanzadas que en etapas tempranas, además del coste de los recursos humanos), hasta problemas de tiempo (el no emplear el tiempo suficiente a su debido momento, implica a la larga emplear el doble para tener que solucionar los problemas).

Es por esto anterior, que el propósito principal de este proyecto consiste en analizar a fondo cada uno de los pasos a seguir y las consideraciones a tener en cuenta con el fin de tener garantía de que se realiza una correcta y adecuada definición de requisitos, todo ello con el principal objetivo de que finalmente obtengamos un proyecto exitoso.

Como se podrá observar a lo largo de este, en un proyecto todo va ligado a la especificación de requisitos software, es decir, dependiendo de la definición de esta, así será el diseño y la calidad

entre otros, por lo que el principal objetivo que se persigue, es la realización de una correcta definición de requisitos software.

Por lo tanto, resumiendo todo lo anterior el objetivo principal de este proyecto es la realización de una correcta definición de la especificación de requisitos software dentro de un proyecto, y en base a ese objetivo principal, se proponen objetivos parciales como que el proyecto cumpla con los estándares de calidad entre otros.

1.3 Fases de desarrollo

Para llevar a cabo el desarrollo de este proyecto, se ha tenido que pasar por diferentes fases que se mencionan a continuación:

- Buscar el conjunto de información necesaria, en un primer lugar para completar yo el conocimiento necesario acerca del tema del proyecto, y en segundo lugar para poder comenzar a redactar. Si bien es cierto, a lo largo de todo el proyecto he tenido que ir buscando información, pero el primer empujón fuerte de adquirir esta la tuve que realizar en la etapa inicial.
- Mucha de la documentación de la que disponía, era en inglés, por lo tuve que traducir muchísima de esta al español, ya que por lo que he podido observar a lo largo del proyecto, la documentación en inglés es muy valiosa.
- Realizar un índice “orientativo” y estructurado para hacer la función de guía. Indico orientativo, porque del índice inicial que hice al índice final que ha resultado hay una gran diferencia.
- A partir de aquí, y durante todos los meses que ha durado el proyecto, ha consistido en ir punto por punto, buscando, leyendo, comprendiendo, en ocasiones traduciendo, preguntando a personas de interés y resumiendo información relacionada con el punto en el que me encontraba con el fin de sacar la mejor esencia.
- Finalmente se han realizado dos casos prácticos, uno relacionado con calidad y programado en HTML y que está relacionado con una encuesta de satisfacción a los agentes, y otro caso práctico en el que se realiza la definición del conjunto de requisitos.

1.4 Medios empleados

En cuanto a los medios empleados para la realización de este proyecto han sido varios, entre los que destacan los siguientes:

- **Internet** como herramienta de búsqueda para información en páginas conocidas y de prestigio, por poner un ejemplo la página de “Inteco”, actualmente “Incibe”.
- **Bibliografía** proveniente en su mayor parte de la biblioteca de la Escuela Politécnica de la Universidad Carlos III.
- **Experiencia propia**, y creo que en mi opinión es la información más valiosa, ya que leer un libro te aporta todos los conceptos necesarios para saber del tema, pero el hecho de participar en pruebas de entornos donde se tienen que cumplir unos ciertos requisitos, te ayuda a entender y vivir la situación real, ya que una cosa es la teoría, y otra la práctica.

- En mi caso, el **tutor** del proyecto ha sido una gran fuente de información, ya que aparte de dar directrices para saber por dónde abordar cada tema, te proporciona información, en este caso como por ejemplo estándares y bibliografía de interés.
- **Mis jefes**, ya que al disponer de muchísima experiencia me han aportado conceptos, ideas y ejemplos con situaciones reales.

1.5 Estructura de la memoria

La memoria consta de casi 200 páginas aproximadamente, donde aunque el tema principal es la definición del conjunto de requisitos software, podemos observar claramente partes diferenciadas.

Con el fin de ayudar a la comprensión de este proyecto fin de carrera, a continuación se muestra un escueto resumen de cada uno de los capítulos que forman este.

Capítulo 2: información general acerca de los proyectos software, que engloba la gestión de estos así como también algunos de los modelos de software existentes.

Capítulo 3: contiene clasificaciones de la ingeniería de requisitos según diferentes autores como son Ian Sommerville y Roger Pressman.

Capítulo 4: engloba distintos aspectos, todos ellos relacionados con los requisitos como son la definición y gestión de requisitos y actividades que tienen relación con los requisitos entre otros.

Capítulo 5: contiene toda la información relacionada con el documento de especificación de requisitos software, no el contenido en sí, sino el conjunto de aspectos relacionados con este.

Capítulo 6: en este capítulo sí que nos encontramos con toda la información pertinente que recoge el documento de especificación de requisitos, esto es, se define cada uno de los apartados según un estándar específico.

Capítulo 7: este séptimo capítulo se centra principalmente en la obtención de requisitos, esto es, desde los métodos de obtención, pasando por los problemas asociados a esto hasta los modelos generales del proceso de obtención y análisis de requisitos.

Capítulo 8: engloba la definición de cada uno de los tipos de requisitos existentes, comenzando desde los requisitos funcionales y finalizando por los requisitos del dominio.

Capítulo 9: toda la información contenida en este noveno capítulo está relacionada con el modelo CMMI, hablando del desarrollo, gestión, análisis y validación de requisitos mediante dicho modelo.

Capítulo 10: este capítulo incluye información adicional de requisitos software, pero en sistemas críticos.

Capítulo 11: engloba todas las actividades relacionadas con la validación de requisitos, es decir, desde la definición de en qué consiste, pasando por las técnicas relacionadas con estas, hasta algunas observaciones de interés relacionadas con esto.

Capítulo 12: todo este capítulo está relacionado con el tema de la calidad, tratando aspectos como tipos de herramientas para ello, normas y estándares relacionados y ya más generales la calidad en productos software.

Capítulo 13: este capítulo está formado por el presupuesto, donde incluye un desglose del coste del proyecto incluyendo recursos materiales, recursos humanos y coste total de este.

Capítulo 14: contiene el glosario, formado este por aquellos términos que considero oportuno conocer o que en ocasiones son complicados de interpretar.

Capítulo 15: contiene las siglas que han sido utilizadas en el proyecto.

Capítulo 16: contiene el conjunto de bibliografía, estándares y en general todas las fuentes utilizadas para la realización del proyecto.

Capítulo 17: este capítulo contempla las principales conclusiones y las posibles líneas futuras en base a las cuales el proyecto puede avanzar.

Capítulo 18: este contiene un anexo que se corresponde con un caso práctico donde se analizan el conjunto de requisitos software para un sistema concreto y modificado del actual, la casación del mercado eléctrico a nivel europeo.

Capítulo 2: Proyectos software

En primer lugar y para centrar el tema del proyecto, se comentará de forma general y resumida en qué consiste un proyecto software así como también las diferentes estimaciones que han de realizarse para el comienzo de este, para que con ello se sepa qué lugar ocupa la fase de requisitos, tema principal de este proyecto.

Existen diferentes definiciones de proyecto software entre ellas se puede definir proyecto software como:

“Un proceso que consume recursos y está sujeto a influencias externas (requisitos que cambian continuamente, costes, plazos y recursos) y/o internas (dificultades técnicas de producción, sobreestimación de productividad entre otras). Dicho proceso tendrá una serie de objetivos tangibles, que normalmente serán entregados en un plazo y con un coste (y posiblemente con un precio) y con unos niveles de calidad, asumiendo por parte de quien lo realiza una serie de riesgos.” [LA98]

En la mayoría de los proyectos software es imprescindible el estudio de diversas soluciones desde diferentes puntos de vista, previamente a la realización de un proyecto software. Dicho procedimiento a seguir se denomina “Análisis de valor” y engloba las actividades siguientes:

- Identificación de Requisitos: en términos generales consiste en adquirir una primera idea del conjunto de requisitos.
- Análisis del producto en su entorno y determinación de restricciones: se analizará el entorno de prueba y las tecnologías de desarrollo entre otros, determinando los requisitos de interfaz y las restricciones (mantenimiento, desarrollo y operación).
- Análisis del ciclo de vida del producto: Se determinarán las fases iniciales del ciclo de vida en términos de duración, operación y desarrollo entre otros. Esta fase da opción a la identificación de nuevos requisitos.
- Estudio de los distintos puntos de vista acerca del producto: estos puntos de vista de los que hablamos son tomados de usuarios que de alguna manera interactúan con el producto.

A continuación, se menciona el conjunto de características detalladas de los proyectos software:

- ❖ Invisibilidad: referido a que en cualquier proyecto que no sea software, el avance de este es visible notablemente, mientras que en el software el avance no se percibe de inmediato. El hecho de que un proyecto software posea invisibilidad, implica dificultad a la hora de realizar el conjunto de estimaciones.
- ❖ Complejidad: debido a que se está hablando en todo momento de proyectos software, el software en sí mismo es mucho más complejo que cualquier otro producto de ingeniería. Entre otros, esto se revela en las pruebas que son realizadas a este, donde una prueba al 100% es imposible ya que no se puede llevar a cabo, mientras que otros productos de ingeniería se prueban al 100%.
- ❖ Flexibilidad: debido a que tanto los requisitos definidos para el proyecto en cuestión como el software es muy simple cambiarlo o modificarlo.

- ❖ Procesos y estrategias de desarrollo variadas y poco probadas y definidas: existen diferentes modos de desarrollar software.

Debido a que la realización de un proyecto es una acción complicada y compleja, requiere una administración del trabajo, ya que en la mayoría de los casos son muchas personas las que participan y por un tiempo largo. Como se comenta, estos proyectos requieren de una administración, y esta es una actividad fundamental para que el proyecto siga su cauce, ya que la tarea de administrar entre otras conlleva planificación, control de personal y monitoreo.

Si bien es cierto, se puede decir que todas las personas involucradas en el proyecto administran en cierto modo, pero en realidad hay distinguidas personas que son las que realizan tareas concretas, como son los siguientes:

- ❖ Ingeniero del software: esta figura dentro del proyecto planifica, monitorea y controla las tareas técnicas, además de administrar sus actividades cotidianas.
- ❖ Gerentes ejecutivos: estos coordinan la interfaz entre la empresa y los profesionales del software.
- ❖ Gerentes del proyecto: estos planifican, monitorean y controlan el trabajo de un grupo de ingenieros de software.

Para la realización de una administración eficaz de un proyecto software se debe poner especial atención en estos cuatro aspectos (las cuatro “P” de la ingeniería del software):

- *Proyecto*: Debido a que no se conoce otra manera de administrar algo tan complejo como es un proyecto software, estos primeramente se planifican y posteriormente se revisan, ambas tareas deben realizarse minuciosamente. Aun así, los resultados no son del todo buenos, ya que existe un porcentaje muy bajo, de un 10%, que son éxitos hablando en términos de tiempo, coste y calidad frente a un 70% que o bien se dan por finalizados sin haber sido acabados o estos presentan retrasos.
- *Proceso*: el proceso aporta el marco conceptual a partir del cual puede fijarse un plan completo con el fin de llevarse a cabo el desarrollo del software. Existen un conjunto de tareas que son comunes a todos los proyectos, independientemente del tamaño, y por otro lado, existen un conjunto de estas que se aplican dependiendo del tipo de proyecto.
- *Personal*: el personal es un aspecto muy importante que hay que cuidar, tanto es así, que se realizó un modelo de madurez (People -CMM) así como en todo momento se analiza la formación de personas motivadas y altamente calificadas.
- *Producto*: como paso previo a la organización de un producto, es necesario fijar aspectos como son objetivos y ámbitos relacionados con este, así como también definir restricciones técnicas, posibles soluciones alternativas y restricciones administrativas. Esta es información básica y de primera necesidad, ya que sin ella no se pueden realizar estimaciones reales ni la calendarización del proyecto entre otros.

La realización de un proyecto software de forma satisfactoria, implica unos requisitos adecuados, donde estos sean desarrollados de forma adecuada, tengan un grado alto de control de cambios y sean comprendidos y documentados adecuadamente. De no ser así, esto llevará a un aumento de los costes del proyecto, retraso en la entrega de este y una calidad del producto no deseada, lo que se traduce en una insatisfacción por parte del usuario/cliente.

En ingeniería del software, y en este documento en particular, se hablará del término “usuario” y el término “cliente”, donde estos se definen de la siguiente manera:

- Usuario será la persona u organización que va a usar el nuevo producto.
- Cliente será la persona u organización que solicita la realización del producto.

2.1 Gestión de proyectos software

Una vez llegados a este punto, profundizaremos sobre las tareas concretas que engloba la administración de un proyecto software. El conjunto de todas estas actividades forma lo que se llama “planificación de proyecto”. Esta planificación de proyecto es un conjunto de actividades que deben realizarse antes del comienzo del proyecto y cuyo objetivo es calcular una estimación del trabajo que se va a realizar, los recursos que se van a necesitar (materiales y personales) y el tiempo que se va a invertir desde el comienzo al fin. Una vez finalizada esta estimación se realiza un calendario de proyecto, donde se especifica cada una de las actividades concretas del proyecto, así como también el responsable de realizar cada una de ellas.

Debido a que este proyecto se fundamenta en *“el análisis de requisitos en el desarrollo del software”*, este apartado en el que nos encontramos aportará una visión global de las diferentes fases de la gestión de proyectos software, buscando como principal objetivo que el lector sepa en qué fases y de qué manera se llevan a cabo actividades relacionadas con los requisitos, ya sea obtención, actualización o mantenimiento de estos.

La gestión de proyectos, es uno de los pilares fundamentales de la ingeniería del software debido a que una no correcta gestión de un proyecto nos puede llevar en un porcentaje muy alto de los casos al fracaso de proyectos.

Esta actividad es llevada a cabo por un grupo de personas, los gestores del software, y estos deben encargarse de ciertas actividades como son:

- Planificar el desarrollo del proyecto.
- Vigilar el trabajo, con el fin de verificar que se está realizando de acuerdo a los estándares requeridos.
- Revisar el avance del proyecto para comprobar que este se está desarrollando conforme al tiempo y costos establecidos para este.

Todos los gestores de proyectos realizan las mismas labores, pero, en concreto los gestores de proyectos software juegan con diferentes aspectos, lo que hace que dicha gestión sea más complicada de lo habitual. Los factores a los que nos referimos son los siguientes:

- Cuando un proyecto es grande, siempre es diferente a todos los proyectos anteriores, y por ello, ¿Qué implica esto? Que los gestores de proyectos aunque dispongan de muchos años de experiencia, nunca son suficientes como para tener la habilidad de anticiparse a posibles problemas.
- Tenemos la desventaja de que el software es intangible, ¿Qué significa esto? que el software no puede ser tocado, lo que conlleva que en el supuesto caso de darse algún problema, los efectos de este, no son visibles de una manera obvia sino que se fían de otros para hacer la documentación oportuna para chequear el progreso.

- No hay procesos del software estándar, es decir, que los procesos del software cambian significativamente entre las organizaciones.

Somos conscientes de que es imposible describir un conjunto de pasos sistemáticos a realizar por el gestor de proyectos, ya que estos serán diferentes dependiendo por una parte del tipo de producto software, y por otra del tipo de organización, pero normalmente, un porcentaje alto de proyectos software tienen un conjunto de actividades en común relativas con el proyecto y que son las siguientes:

❖ **Estimación**

La estimación se realiza en las primeras fases del ciclo de vida de desarrollo del software, y consiste principalmente en realizar una evaluación del número de personas y tiempo necesario de cada una de ellas, así como también la duración del proyecto y por último el conjunto de recursos necesarios. Debido a que este análisis se realiza como hemos comentado en fases muy tempranas, es necesario estudiar los riesgos asociados a las estimaciones anteriores.

❖ **Organización**

Como se puede observar, es una actividad que se realiza entre la estimación y la planificación de un proyecto. Las principales tareas que se llevan a cabo en la organización del proyecto son:

- ✚ Dividir el proyecto en grupos de actividades básicas con el fin de construir el WBS (Work Breakdown Structure).
- ✚ Identificar los hitos y entregables referentes al proyecto, correspondiéndose con el OBS (Object Breakdown Structure).
- ✚ Formar la estructura de los distintos equipos que realizarán cada una de las tareas (RBS, Resource Breakdown Structure).
- ✚ Repartir el presupuesto del que se dispone entre el conjunto de fases y actividades.

❖ **Planificación**

Un proyecto está formado por un conjunto de actividades, las cuales deben ser realizadas en un determinado orden, y siempre teniendo en mente el conjunto de recursos humanos disponibles y las restricciones técnicas.

En la fase de planificación, principalmente se realiza la asignación de tiempos a cada una de las actividades. Esta tarea, igual que pasa por ejemplo con la organización, se lleva a cabo en varios momentos del ciclo de vida.

❖ **Plan de desarrollo software**

Este documento, en ocasiones nombrado como “PDS” se realiza al principio del proyecto junto con el plan de calidad software (PCS). El principal objetivo del PDS es establecer desde el principio las actividades que dan lugar al proyecto, las características de estas así como también el orden en el que tienen que ser ejecutadas para garantizar el correcto desarrollo del software.

El plan de desarrollo software va dirigido a un conjunto de personas, como son el jefe de proyecto, el responsable de la planificación y seguimiento así como también el que se encarga de la

administración de los recursos, al cliente, que es una de las partes más importantes, ya que este es el encargado de aprobar dicho plan y por último al responsable de calidad.

Para elaborar este documento, debemos de tener a nuestro alcance un conjunto de información específica, que incluye requisitos de calidad incluidos en el PCS, documentos que especifiquen el software que hay que realizar, y por último cláusulas contractuales que especifiquen los requisitos de gestión del proyecto.

❖ **Seguimiento**

Es de vital importancia señalar que las actividades de seguimiento tienen su nacimiento en las fases tempranas del proyecto, y duran hasta la finalización del mismo.

Comprender cuál es el objetivo del seguimiento, es sencillo y consiste en realizar una comparación con el fin de encontrar diferencias significativas entre el plan de acción previsto (se obtiene de las fases de organización y planificación dirigidas por la dirección del proyecto software) y el desarrollo efectivo del proyecto. Entre otros objetivos que se persiguen encontramos lo siguiente:

- Valoraciones regulares con el fin de revisar el avance del proyecto.
- Ejecución del plan de acción para tener en cuenta las desigualdades encontradas.
- Para cada actividad, contabilizar sus costes.
- Predecir los posibles desvíos de importancia con el fin de evitarlos en la medida de lo posible.
- Colaborar para la elaboración de un histórico de proyecto.

Como conclusión a todo lo visto anteriormente acerca de la gestión de proyectos software y relacionado con la actividad de requisitos software, se puede decir que la obtención de requisitos es una de las primeras actividades que se lleva a cabo, no obstante aunque en lo anterior no quede demasiado claro, en los capítulos posteriores de este proyecto, se observará que esto es cierto, ya que es el pilar fundamental para comenzar un proyecto, y que además, dependiendo del ciclo de vida elegido todo lo relacionado con esto como la creación de nuevos requisitos, modificación de los ya existentes o revisión de estos, dependerá de ello.

2.2 Modelos del proceso del software

Uno de los puntos más importantes que determina si un proyecto software tendrá éxito es la correcta elección del modelo del proceso para elaborar el software. Hay que tener en cuenta que dicha elección irá en función de las características del proyecto.

En primer lugar, tenemos que saber que un modelo de proceso es un conjunto de actividades, que ejecutadas, se obtiene como resultado un producto software. En ocasiones, dicho producto software es creado desde cero, pero la mayoría de las veces, el desarrollo del software se realiza transformando y ampliando el ya existente.

Aunque como estudiaremos posteriormente, existen distintos procesos de software, pero todos ellos tienen un conjunto de actividades comunes y primordiales, y que son las siguientes:

- ❖ **Especificación del software**: definición de restricciones y funcionalidad del software.

Análisis de requisitos en el desarrollo del software

- ❖ Diseño e implementación del software: se debe realizar un diseño que satisfaga el conjunto de requisitos definidos en el paso anterior.
- ❖ Validación del software: consiste en validar que el software actúa de una manera adecuada, esto es, que el sistema hace lo que el cliente quiere.
- ❖ Evolución del software: debido a los posibles cambios requeridos por el cliente, el software debe tener la capacidad de progresar con el fin de cubrir las necesidades de estos.

Aun no siendo este el tema principal del proyecto, a continuación se da una pequeña descripción de algunos de los tipos de modelos existentes así como su representación gráfica, ya que indirectamente esto afecta a los requisitos software porque en función del modelo de ciclo de vida elegido para elaborar el software, las cuatro actividades anteriores se organizan de distinta forma y por consiguiente los requisitos se tratarán de una manera u otra.

<u>MODELO DE CICLO DE VIDA</u>	<u>DESCRIPCION</u>
Code-and-fix	Se suele utilizar para proyectos pequeños o en prototipos para estudiar la viabilidad técnica. Consiste en programar sin pensar en requisitos, ni en tiempo ni tampoco en actividades relacionadas para programar.

Tabla 1: Modelo de ciclo de vida Code-and-fix

<u>MODELO DE CICLO DE VIDA</u>	<u>DESCRIPCION</u>
Waterfall (Cascada)	<p>Es una mejora de la técnica anterior, y consiste en la ejecución temprana de tareas de definición (requisitos) y documentación, esto previo a la codificación. En este caso de modelo de ciclo de vida, cada actividad tiene como requisito estar finalizada para poder pasar a la actividad siguiente, por lo que esto genera gran rigidez en el desarrollo. El inconveniente que se encuentra es que los errores son encontrados en fases ya muy avanzadas. Dicho modelo está formado por diferentes etapas, las más generales se describen a continuación:</p> <ul style="list-style-type: none">➤ <u>Análisis y definición de requisitos</u>: a través de la información aportada por los usuarios se definen aspectos como restricciones, servicios y metas referentes al sistema.➤ <u>Diseño del sistema y del software</u>: los requisitos obtenidos en el paso anterior, se clasifican en hardware y software y por consiguiente se determina la arquitectura completa del sistema.➤ <u>Implementación y pruebas de unidades</u>: en esta fase se realiza el diseño del software, este se lleva a cabo como unidades de programa.➤ <u>Integración y prueba del sistema</u>: cada una de las unidades individuales se unen y se prueban simulando el sistema completo con el fin de cerciorarnos que se cumplen los requisitos software.➤ <u>Funcionamiento y mantenimiento</u>: es la fase más larga de todas, y el

	objetivo del mantenimiento consiste en subsanar los errores que no han sido encontrados en las etapas previas del ciclo de vida.
--	--

Tabla 2: Modelo de ciclo de vida en cascada

Este modelo de forma gráfica se puede representar de la siguiente manera:

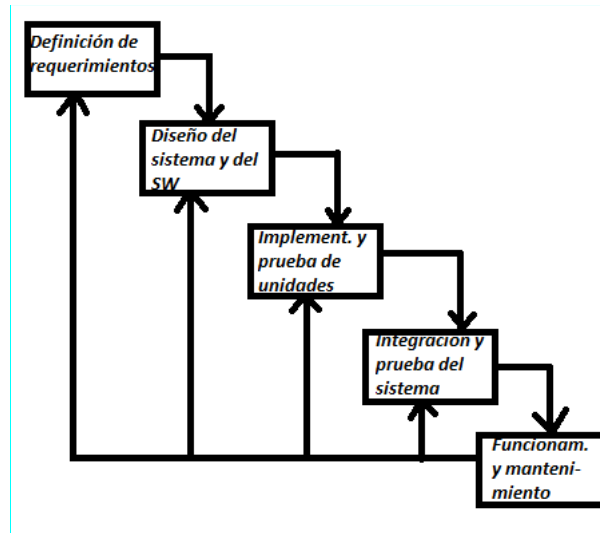


Ilustración 2: Esquema del ciclo de vida en cascada, tomado [Som05]

<u>MODELO DE CICLO DE VIDA</u>	<u>DESCRIPCION</u>
En "V"	Este modelo de ciclo de vida está formado por un conjunto de fases, en las que cada una de ellas a su vez la forman un conjunto de actividades. Este tipo de descomposición garantiza la viabilidad, coordinación, estimación y seguimiento de las actividades y por consiguiente la localización de desviaciones y la ejecución de las acciones pertinentes con el fin de subsanar el problema.

Tabla 3: Modelo de ciclo de vida en "V"

A continuación se muestra gráficamente, para que se entienda mejor el modelo y el porqué de su nombre:

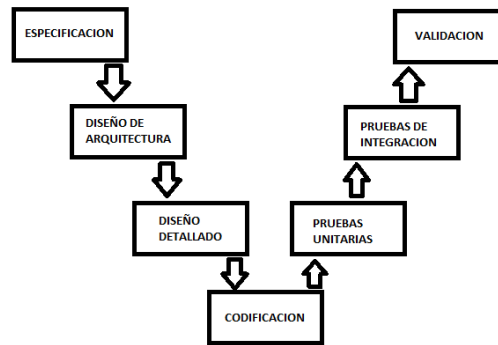


Ilustración 3: Esquema del ciclo de vida en "V"

<u>MODELO DE CICLO DE VIDA</u>	<u>DESCRIPCION</u>
Espiral	El principal objetivo de este modelo es la reducción de riesgo y es caracterizado por el estudio de alternativas. Como su nombre indica, se representa mediante una espiral, en la que el radio de esta representa el esfuerzo empleado y el recorrido angular simboliza el progreso. A grandes rasgos, en cada vuelta dada a la espiral se realiza por una parte el establecimiento de propósitos, posibles opciones y restricciones, valoración de posibles alternativas y reconocimiento y solución de riesgos así como también desarrollo del siguiente nivel de producto y programación del paso siguiente. La gran ventaja del uso de este modelo es el grado de adaptabilidad que tiene.

Tabla 4: Modelo de ciclo de vida en espiral

<u>MODELO DE CICLO DE VIDA</u>	<u>DESCRIPCION</u>
Prototipado	Este modelo puede ser usado a lo largo del ciclo de vida y le da opción al cliente a analizar y observar posibles alternativas. Esto es una ventaja, ya que no siempre dicho cliente puede manifestar el conjunto de requisitos con gran detalle, siendo de necesidad enseñarle posibles alternativas a su problema, con la ventaja del ahorro posteriormente de tiempo y recursos. Este modelo minimiza costes y plazos debido a que reconoce problemas de requisitos. Tener en cuenta que este modelo no debe usarse para la construcción de un sistema final, únicamente si se usa la técnica de prototipos evolutivos.

Tabla 5: Modelo de ciclo de vida "prototipado"

<u>MODELO DE CICLO DE VIDA</u>	<u>DESCRIPCION</u>
Entrega incremental	<p>Este tipo de modelo compromete a que el desarrollo del software se realice en grupos de capacidades funcionales. Además en la planificación, deben definirse tamaño, número y etapas de construcciones incrementales. Algunas ventajas que aporta este modelo son las siguientes:</p> <ul style="list-style-type: none">- Los usuarios pueden sacar provecho del sistema sin tener que esperar a que este esté completo.- El porcentaje de fallo total del proyecto es muy bajo, aunque no se descarta la aparición de problemas en ciertos incrementos.- Los incrementos iniciales pueden ser usados como prototipos y de esta manera conseguir experiencia en relación con los requisitos de los incrementos posteriores del sistema.

Tabla 6: Modelo de ciclo de vida "entrega incremental"

De forma gráfica se puede visualizar de la siguiente manera:

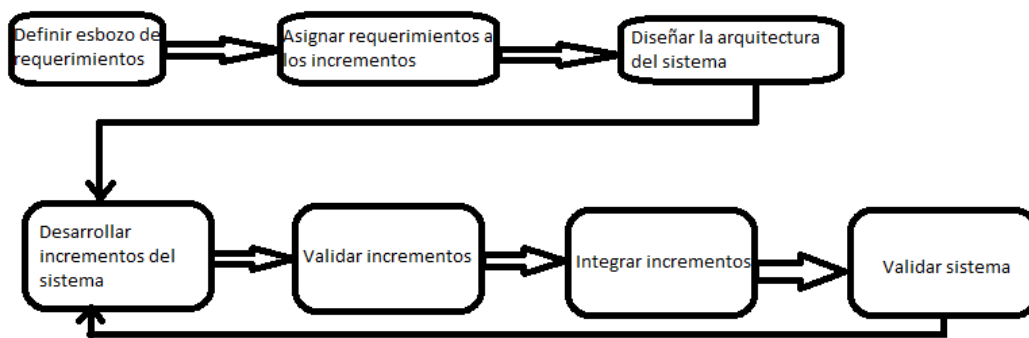


Ilustración 4: Esquema del ciclo de vida "entrega incremental" tomado [Som05]

Capítulo 3: Ingeniería de requisitos

El término “Ingeniería de requisitos” ha sido creado para englobar el conjunto de actividades relacionadas con los requisitos, como son descubrimiento, documentación y mantenimientos del conjunto de estos.

El proceso de la ingeniería de requisitos, es un conjunto ordenado de actividades que se siguen, cuyo principal objetivo es validar y mantener el documento de requisitos del sistema. Cabe mencionar, que una especificación íntegra del proceso debe contener información relativa al conjunto de actividades que se van a realizar, la programación de estas actividades así como también la persona responsable de cada una de ellas, las herramientas a utilizar y las entradas y salidas de cada una de las actividades.

Es de gran importancia que las organizaciones dispongan de un proceso de ingeniería de requisitos preciso y concreto, pero pocas disponen de ello. En las organizaciones en general, las personas implicadas en el proceso, son las encargadas de tomar las decisiones acerca de qué hacer, como hacerlo e identificar las herramientas a utilizar. La organización únicamente se encarga de definir los resultados del proceso.

Una correcta definición del proceso de ingeniería de requisitos, es beneficioso para la organización, ya que de esta manera cada una de las personas involucradas en dicho proceso dispondrán de esta a modo de guía, y por otra parte, el conjunto total de actividades serán llevadas a cabo, ya que la probabilidad de olvido es menor que si no se dispone de ello.

La ingeniería de requisitos abarca por una parte la definición de los servicios, es decir la especificación de cada una de las necesidades de los clientes y por otra parte las restricciones, todo ello a tener en cuenta a la hora del diseño y la implementación del sistema. Todos y cada uno de estos aspectos estarán recogidos en un único documento. Este documento al que nos referimos, es un documento oficial y de carácter importante para usuarios finales, clientes y desarrolladores de software entre otros. Dependiendo de la organización, este será nombrado de distintas formas, como por ejemplo: definición de requisitos, especificación funcional o especificación de requisitos software (SRS). A lo largo del proyecto, siempre que se tenga que hacer algún tipo de referencia a este documento, será nombrado como “Documento de especificación de requisitos software”.

Dicho documento, es uno de los primeros pasos que se realiza en un proyecto, y en mi opinión de gran importancia, ya que los posibles errores que se cometan en este momento, serán arrastrados a lo largo de todo el proyecto. Recalco que es de gran importancia, debido a que este será la base de los pasos sucesores, ya que en todo momento el principal objetivo que se tiene en mente es el cumplimiento del conjunto de requisitos, ya que a groso modo, estos reflejan todas y cada una de las exigencias del cliente.

A continuación se mencionan las diferentes fases que engloban la ingeniería de requisitos. Dependiendo del desarrollador, estas fases variarían ligeramente de una clasificación a otra, por lo tanto seguidamente se muestra un cuadro resumen en el que se puede encontrar las diferentes clasificaciones para Ian Sommerville y Roger S. Pressman .

	Ian Sommerville	Roger S. Pressman
Actividades	Estudio de viabilidad	Concepción
	Obtención y análisis de requisitos	Indagación
	Especificación de requisitos	Elaboración
	Validación de requisitos	Negociación
		Especificación
		Validación
		Administración de los requisitos.

Tabla 7: Cuadro comparativo según los autores Ian Sommerville y Roger Pressman

Llegados a este punto y una vez que se tiene una visión general de las diferentes clasificaciones, se explicará en detalle cada una de sus fases, para comprender que se hace en cada una de ellas y cuál es su principal objetivo.

3.1 Clasificación según Ian Sommerville

Según este desarrollador y como se puede observar en el cuadro resumen, clasifica las actividades de la ingeniería de requisitos en cuatro fases: estudio de viabilidad, obtención y análisis de requisitos, especificación de requisitos y validación de requisitos. A continuación se tiene el detalle de cada una de ellas, adaptado de [Som05]:

- Estudio de viabilidad: en esta primera fase se estudia si las necesidades del cliente se pueden satisfacer por una parte con el presupuesto con el que se dispone para dicho proyecto, y por otra con las tecnologías existentes en el mercado. Además de analizar si realmente dicho proyecto es beneficioso desde el punto de vista del negocio. Como conclusión se informa si posteriormente se seguirá con el estudio.
- Obtención y análisis de requisitos: esta es la fase de obtención de cada uno de los requisitos. Estos requisitos podrán ser obtenidos por diferentes fuentes, como por ejemplo los futuros usuarios del sistema, mediante la observación de otros sistemas similares...etcétera. El fin de toda la información recogida de las diversas fuentes, será la creación de uno o varios prototipos, los cuales ayudarán a comprender el sistema.

- Especificación de requisitos: llegados a este punto, la información recopilada en el paso anterior, es decir en la fase de obtención y análisis de requisitos, será plasmada en el documento de especificación de requisitos software, donde cada uno de los requisitos será clasificado dependiendo de unos criterios determinados.
- Validación de requisitos: en esta fase se verifica que toda la información contenida en el documento de especificación de requisitos software es real, completa y consistente. En este punto se encontrarán errores, los cuales deben ser corregidos.

Se ha de tener en cuenta que las cuatro fases mencionadas anteriormente, no tienen por qué realizarse en dicho orden, ya que puede haber solapamiento entre ellas.

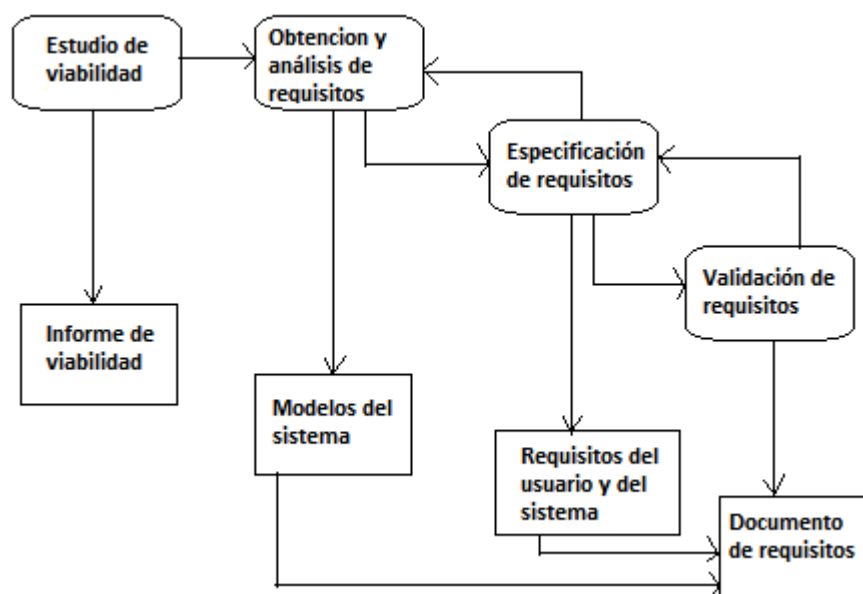


Ilustración 5: Actividades de la ingeniería de requisitos tomado [Som05]

3.1.1 Estudios de viabilidad

El estudio de viabilidad es la primera fase y debe comprender un periodo corto de tiempo, aproximadamente tres semanas, en el que el principal objetivo de dicho estudio, es llegar a saber con certeza, si realmente es conveniente seguir con el proceso de desarrollo del sistema, y en caso afirmativo realizar un informe con las diversas propuestas.

Para llegar a saber si es viable el desarrollo del sistema, se precisa obtener información minuciosa y detallada sobre el sistema a desarrollar. Dicha información, se puede obtener de diversas fuentes, como por ejemplo expertos en tecnología, jefes de departamentos donde va a ser utilizado el sistema, usuarios finales etcétera, todo ello, para llegar a resolver varias cuestiones claves de gran importancia, como son algunas de las siguientes:

- ✓ ¿Favorece el sistema a desarrollar a los objetivos generales de la organización?

- ✓ ¿Se puede implementar el sistema utilizando la tecnología actual disponible en el mercado?
- ✓ ¿Dicha implementación se puede llevar a cabo con el presupuesto del que se dispone? ¿Y también en el periodo de tiempo que se propone?
- ✓ ¿Puede integrarse el sistema con otros sistemas ya existentes en la organización?

Una vez que toda esta información ha sido recopilada, se debe realizar el informe correspondiente con las conclusiones a las que se ha llegado, sugiriendo los posibles cambios a realizar.

3.1.2 Obtención y análisis de requisitos

Como comentario adicional y antes de explicar en detalle el paso de “obtención y análisis de requisitos”, es de carácter importante mencionar a los “stakeholders”. Cuando se habla de stakeholders, se refiere al conjunto de personas que de algún modo se verán afectadas por el sistema y las cuales tienen influencia directa o indirectamente en los requisitos de este. Este conjunto de personas engloban a los ingenieros responsables del desarrollo y mantenimiento del sistema, usuarios finales y organismos externos tales como autoridades de certificación entre otros. En el proceso de ingeniería de requisitos, es aconsejable realizar un listado de “stakeholders” en las etapas tempranas.

Una vez citados los “stakeholders”, continuamos con la segunda fase, la obtención y análisis de requisitos. En esta fase, la persona encargada de obtener toda la información necesaria a través de los diferentes medios, es el ingeniero de software, trabajando este junto con el cliente y el usuario final del sistema. Dicho ingeniero será el encargado de indagar para saber cuáles son los servicios que tiene que prestar el sistema, así como también las restricciones de este entre otros.

Las principales actividades del proceso de obtención y análisis de requisitos, se detallan a continuación:

i. Descubrimiento de requisitos

En esta actividad, mediante diferentes fuentes como los stakeholders, la documentación existente y mediante sistemas similares ya presentes, se recolecta gran cantidad de información con el fin de obtener los requisitos de usuario y los requisitos del sistema.

Para obtener la información proveniente de los stakeholders, se pueden realizar entrevistas, observar e incluso utilizar diferentes escenarios y prototipos para llegar a obtener mayor número de requisitos.

A continuación, se resumen brevemente algunas de las técnicas existentes que pueden ser utilizadas con los stakeholders:

- ❖ Entrevistas: Es la técnica más utilizada, y consiste en un juego de preguntas que realiza un entrevistador a un usuario o grupo de usuarios del sistema. Para la realización de una entrevista, se requiere una gran preparación anterior a esta por parte del entrevistador ya que conlleva el análisis de las preguntas a realizar y las personas a entrevistar entre otras.

- ❖ Observación: Consiste en observar el lugar o departamento que se quiere informatizar con el nuevo sistema. La observación tiene ciertas ventajas con respecto a las otras técnicas, ya que esta, al estar la propia persona observando, se llega a examinar todo el funcionamiento real de la empresa y todos y cada uno de los detalles.
- ❖ Prototipado: Consiste en la fabricación de una maqueta del sistema, en la que los usuarios de dicho sistema, pueden observar de una mejor manera, si este cumple las necesidades para el que ha sido construido.
- ❖ Cuestionarios: es una técnica muy eficaz, ya que permite en un periodo corto de tiempo, obtener la información deseada de un gran número de personas. Los cuestionarios tienen que tener una serie de características como son: preguntas breves y concisas, no ambigüedad, redacción clara, preguntas centradas en el tema que queremos informarnos y además una breve introducción que indique el propósito del cuestionario y unos agradecimientos por la colaboración.
- ❖ Estudio de documentación: Todas las empresas disponen de documentación donde se encuentra el funcionamiento y normas de la empresa. Luego el analista debe de analizar minuciosamente dichos documentos para hacerse una idea de las normas de la empresa.

ii. Clasificación y organización de requisitos

En esta actividad, se recopilan todos los requisitos que se han recogido en la actividad anterior y se clasifican según unos criterios determinados.

iii. Ordenación por prioridades y negociación de requisitos

Debido a que la información se recopila de varias fuentes, y entre ellas de muchos stakeholders, en muchos de los casos, algunos de los requisitos entran en conflicto, luego en esta actividad se dará prioridad a los requisitos y se solucionarán dichos conflictos mediante negociación.

iv. Documentación de requisitos

Se documentan todos los requisitos y se crean documentos, ya sean formales o informales.

3.1.3 Especificación de requisitos

En esta tercera fase, toda la información recogida en los pasos anteriores se convierte en un conjunto de requisitos, clasificados estos siguiendo unas determinadas pautas y obteniendo así el documento formal, llamado documento de especificación de requisitos software. Se ha de tener en cuenta que el documento de especificación de requisitos software (ERS), sería deseable que tuviera una serie de características, con el fin de que este sea considerado como una especificación de calidad. Dichas características son las siguientes:

✓ Correcta

Una especificación de requisitos software dispondrá de esta característica, si cada uno de los requisitos definidos en dicha especificación se encuentra en el software.

Si buscamos la definición de “correcta” en el ámbito de ingeniería del software, según el estándar IEE 830, 1998 nos encontramos con lo siguiente: *“Un conjunto de requisitos es correcto solo si todos los requisitos contenidos representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño”*.

Se ha de tener en cuenta, que la realización de una especificación de requisitos correcta depende principalmente del usuario final del sistema, es decir, la persona encargada de determinar si la especificación de requisitos software posee esta característica, es el cliente el cual demanda el sistema. Es por esto, que en la especificación de requisitos software se debe comprobar si satisface dicha característica a través de lo siguiente:

- Verificación y aprobación del usuario.
- Problemas comunes en este entorno como exclusión de información significativa proveniente de los usuarios y la suma de requisitos no demandados, estos provenientes de los equipos de análisis y desarrollo.

✓ No ambigua

La descripción de cada uno de los requisitos software, debe de llevar a una única interpretación.

✓ Comprobable

Se dice que un requisito es comprobable si este puede ser validado, en caso contrario, si este no puede ser validado, por regla general tendrá poco valor práctico. Esta verificación de la que se habla, es fácil, bastará con comprobar que existe un procedimiento determinado y efectivo en coste para que un individuo o máquina verifique que dicho requisito se cumple en el software creado.

✓ Completa

Debe tenerse en cuenta todo lo relativo al rendimiento, funcionalidad y restricciones entre otros. Además, destacar que es necesario que todo esté apropiadamente documentado, ya que no se debe de dejar ningún aspecto para estudiar posteriormente en el proyecto.

El hecho de que una especificación de requisitos sea completa, garantiza que no existen exclusiones que comprometan los requisitos establecidos.

La especificación de requisitos software, será completa si cumple cada uno de los siguientes aspectos:

- Contiene la totalidad de los requisitos significativos del software.
- Especifica todas las respuestas ante todas las posibles entradas.
- Está conforme con cualquier estándar de especificación que se deba cumplir.
- Están etiquetadas y referenciadas en el texto todas las figuras, tablas y diagramas.

✓ Inequívoca

Es necesario que la especificación de requisitos contenga un conjunto de requisitos definidos adecuadamente y que estos aporten una idea definida del futuro sistema.

✓ Fácil de verificar

Es fácil de verificar cuando para cualquier requisito del documento, se puede comprobar que este se cumple satisfactoriamente.

✓ Consistente

Para que la especificación de requisitos sea consistente, debe tener un conjunto de características. Algunas de ellas se mencionan a continuación:

- El documento será consistente cuando ningún requisito entre en conflicto o se contradiga con otro requisito o grupo de requisitos.
- El documento será consistente cuando el conjunto de requisitos sean fáciles de leer y comprender, así como también su escritura debe ser sencilla y clara por si se da el caso de que en un futuro puede ser utilizada por los usuarios.

Hay que tener en cuenta que cuanto más amplio es el número de requisitos, la consistencia es más difícil de conseguir.

✓ Clasificada por importancia o estabilidad

Los requisitos tienen que tener un orden de prioridad fijado, en base a la importancia que tienen cada uno de ellos, o bien una clasificación de la estabilidad, basada en la resistencia a la volatilidad. Este tipo de orden implica que los desarrolladores puedan tomar mejores decisiones acerca de qué requisitos pueden quedarse.

Existen dos tipos de grados:

➤ *Grado de estabilidad*

Valor referente al número de modificaciones en los requisitos que se esperan. Dichas modificaciones se basan en usuarios y métodos que respaldan el sistema del software.

➤ *Grado de necesidad*

Se pueden diferenciar los requisitos en tres categorías, las cuales se mencionan a continuación:

- *Esenciales*: factor de gran utilidad que tendría que tenerse en cuenta si el tiempo referente al proyecto empieza a ser urgente.
- *Condicionales*: se refieren a requisitos que reforzarían al producto del software.
- *Opcionales*: formado por un conjunto de funciones en una clase, las cuales pueden valer la pena o por el contrario no ser redundantes.

✓ Fácil de modificar

Será fácil de modificar si cualquier cambio necesario en los requisitos del documento, se pueda realizar fácilmente y sin ningún tipo de problema.

✓ Fácil identificación del origen y de las consecuencias de cada requisito

Partimos de la base de que cada requisito es necesario que esté identificado, ya sea con un número, secuencia alfanumérica o con alguna letra.

Por otra parte, debido a que serán usados determinados términos para explicar los aspectos a tener en cuenta en la identificación, a continuación se describen los dos tipos de rastreabilidad existentes:

- Referencias hacia delante: depende de cada requisito de la especificación de requisitos el que tenga un nombre o un número de referencia único que sirva para identificarlo en futuros documentos.
- Referencias hacia atrás: Depende de que los requisitos referencien explícitamente sus fuentes en documentos previos.

Existe un conjunto de aspectos de gran importancia a tener en cuenta en la identificación del origen y de las consecuencias de cada requisito, estos son mencionados a continuación:

- Cuando el código referente al requisito es modificado, así como también los documentos, es fundamental revisar el resto de requisitos que pueden verse afectados por dichas modificaciones.
- Las referencias hacia delante de la especificación de requisitos software, son altamente importantes para la conservación del software.
- Deben ser proporcionadas las referencias tanto hacia delante como hacia atrás en el ciclo de vida, cuando un requisito correspondiente a la especificación de requisitos constituye un desglose o una bifurcación de otro requisito.

- ✓ De fácil utilización durante la fase de explotación y de mantenimiento:

Dependiendo del sistema futuro que se vaya a desarrollar, el documento de especificación de requisitos software será realizado con mayor o menor detalle. Se debe tener en cuenta, que dicho documento es utilizado por diferentes personas y para diferentes usos, algunos de estos usuarios son los siguientes:

- *Clientes del sistema*: estos detallan cada uno de los requisitos que desean para el sistema a desarrollar, además de leer y verificar que dichos requisitos cumplen sus exigencias. Estos usuarios serán los responsables de indicar los cambios a realizar en los requisitos.
- *Administradores*: usan el documento de requisitos para planificar una oferta por el sistema y para planificar el proceso de desarrollo del sistema.
- *Ingenieros de sistema*: usan los requisitos para la comprensión del sistema a desarrollar.
- *Ingenieros probadores del sistema*: debido a que el sistema desarrollado debe validarse, estos ingenieros usan los requisitos para el desarrollo de todas las pruebas que componen dicho proceso de validación.
- *Ingenieros encargados del mantenimiento del sistema*: usan los requisitos para la comprensión del sistema y la relación que existe entre cada una de las partes de este.

La estructura del documento de especificación de requisitos se muestra a continuación, hay que tener en cuenta que no es necesario que dicho documento tenga todos y cada uno de los apartados, pero sí señalar que para una correcta y completa especificación de requisitos software, dicho documento debería contener toda la información que en apartados posteriores se detalla.

3.1.4 Validación de requisitos

En esta cuarta y última fase será donde se compruebe que el conjunto de requisitos son correctos, es decir, que estos definen de manera correcta el sistema y satisfacen las necesidades que el cliente desea implica a diseñadores del sistema, ingenieros de requisitos y a los actores del sistema.

Nos encontramos ante una fase de gran importancia, ya que un error en el documento de requisitos y no detectado en este punto, puede llegar a tener fuertes consecuencias de aumento de costos, ya que una vez que es descubierto en otra etapa, ya sea durante el desarrollo o incluso cuando el sistema esté en uso, acarrea el tener que repetir gran cantidad de trabajo, lo que implica un aumento de presupuesto y además un aumento de tiempo, ya que el sistema debe de ser probado de nuevo.

Durante la fase de validación de requisitos, se realiza un conjunto de comprobaciones en el documento de especificación de requisitos, las cuales se indican a continuación:

1. *Verificaciones de validez.* Un usuario puede pensar que se necesita un sistema para llevar a cabo ciertas funciones. Sin embargo, el razonamiento y el análisis pueden identificar que se requieren funciones adicionales o diferentes. Los sistemas tienen diversos stakeholders con diferentes necesidades, y cualquier conjunto de requisitos es inevitablemente un compromiso en el entorno del Stakeholder.
2. *Verificaciones de consistencia.* Comprobar que los requisitos de dicho documento no se contradicen, es decir no se puede dar el caso, que para una misma funcionalidad del sistema, haya dos requisitos contrapuestos.
3. *Verificaciones de completitud.* Verificar que el documento de requisitos contempla todas las tareas y limitaciones sugeridas por el usuario del sistema para este.
4. *Verificaciones de realismo.* Comprobar que cada uno de los requisitos puede ser implementado utilizando la tecnología existente en el mercado, teniendo en cuenta el presupuesto y las limitaciones de tiempo para el desarrollo del sistema.
5. *Verificabilidad.* Comprobar que cada uno de los requisitos existentes en el documento de especificación de requisitos, es verificable, es decir, que con determinadas pruebas al sistema, se puede comprobar que el requisito se cumple, de esta manera se reducen los posibles conflictos entre el cliente y el contratista.

Aunque todas las comprobaciones anteriormente descritas se realizan, prácticamente nunca se encuentran todos los errores en la validación de requisitos, luego es irremediable que haya cambios adicionales de requisitos para corregir las omisiones y las malas interpretaciones después de que el documento de especificación de requisitos haya sido aprobado.

3.2 Clasificación según Roger Pressman

Según Roger S. Pressman la actividad de la ingeniería de requisitos, la resume en siete fases, las cuales son concepción, indagación, elaboración, negociación, especificación, validación y administración de los requisitos. A continuación se define detalladamente cada una de ellas tomadas de [Pre10].

3.2.1 Concepción

Realmente no existe una forma única de iniciar un proyecto software, en algunas ocasiones solo basta con una simple conversación y en la mayoría de los casos, se inicia un proyecto cuando se detecta una necesidad de negocio. Los altos cargos del negocio realizan diferentes acciones de gran importancia, como son:

- Definen caso de negocio para la idea.
- Identifican el ritmo y profundidad de mercado.
- Análisis de factibilidad.
- Identificación de una descripción funcional del alcance del proyecto.

El conjunto de esta información es cambiante, pero en principio es lo justo y necesario para analizar el caso con la organización de ingeniería del software.

En esta fase se define las personas que optan por una solución, el entendimiento básico del problema, la naturaleza de la solución que se quiere, además de la eficacia de la comunicación y colaboración entre el equipo de software y los otros participantes.

3.2.2 Indagación

La fase de indagación consiste en obtener información relevante acerca del principal objetivo del sistema, de cómo va a satisfacer las necesidades del negocio, de cómo va a usarse en las distintas operaciones cotidianas entre otras. Dicho de esta manera parece muy sencillo obtener esta información mediante la realización de preguntas a clientes y usuarios, pero en realidad es una tarea muy complicada de llevar a cabo.

Algunos de los problemas identificados por Christel y Kang [Cri92] en la fase de indagación son:

- Problemas de alcance: Problema debido a que la información aportada por los clientes en lugar de aclarar, se limita a confundir o bien debido a que la frontera de los sistemas no está bien definida.
- Problemas de entendimiento: estos son debido a que los clientes o usuarios del sistema tienen problemas para comunicar sus necesidades, o bien dichos problemas son causados por el hecho de que el cliente no comprende su capacidad y limitación de su ambiente de computación y no saben realmente cuál es su necesidad o bien omitiendo información relevante, la cual ellos no la catalogan como información importante.
- Problemas de volatilidad: Problema debido a que según pasa el tiempo, los requisitos van cambiando respecto a la descripción inicial.

3.2.3 Elaboración

La información recopilada a través del cliente durante las dos fases anteriores se modifica con el objetivo de extender y depurar estadurante la fase de elaboración. El principal objetivo de esta fase es el desarrollo de un modelo de requisitos en el cual se reconozcan los diferentes aspectos del comportamiento, función e información del software.

Aparte de las acciones mencionadas anteriormente, también son realizadas algunas otras como creación y perfeccionamiento de escenarios de usuario que detallen como interactúa el cliente final con el sistema, definición de atributos de cada clase de análisis, identificación de servicios que requieren cada una de ellas e identificación de relaciones y colaboración entre clases, produciendo así diversos diagramas adicionales.

3.2.4 Negociación

En numerosas ocasiones, clientes y usuarios solicitan más de lo que se puede obtener según sus recursos limitados de negocio, o también estos sugieren incluir ciertos requisitos conflictivos, objetando que estos son esenciales para sus necesidades.

Todo este conjunto de cuestiones debe solucionarse mediante un proceso de negociación. Para ello, se solicita a los usuarios del sistema que prioricen sus requisitos y estudien los conflictos.

3.2.5 Especificación

En numerosas ocasiones, clientes y usuarios solicitan más de lo que se puede obtener según sus recursos limitados de negocio, o también estos sugieren incluir ciertos requisitos conflictivos, objetando que estos son esenciales para sus necesidades.

Todo este conjunto de cuestiones debe solucionarse mediante un proceso de negociación. Para ello, se solicita a los usuarios del sistema que prioricen sus requisitos y estudien los conflictos.

3.2.6 Validación

La validación es la fase en la que se comprueba la calidad del producto que se ha originado como fruto de la ingeniería de requisitos. En dicha fase es revisado el documento de especificación de requisitos con el fin de comprobar y verificar que cada uno de los requisitos es cumplido en base a los estándares inicialmente definidos para el producto, el proceso y el proyecto, así como también verificar que el conjunto de requisitos están definidos correctamente, sin ambigüedades e inconsistencias.

Existe un equipo de personas encargadas de realizar la revisión técnica del producto, que es el mecanismo de validación utilizado, donde su principal función es la búsqueda de posibles errores, el análisis de la información para verificar que esta es completa y no hace falta realizar aclaraciones y la modificación de requisitos en caso de encontrarse en conflictos entre ellos. El equipo de personas encargadas de realizar la revisión técnica, está formado por ingenieros del software, usuarios y clientes entre otros participantes.

A continuación se detallan un conjunto de preguntas, las cuales son útiles para el análisis de cada requisito:

- ¿Los requisitos están definidos con claridad? ¿Podrían llegar a tener una interpretación errónea?
- ¿El requisito está acotado en términos cuantitativos?
- ¿Qué otros requisitos se relacionan con este? ¿Están comparados con claridad por medio de una matriz de referencia cruzada u otro mecanismo?
- ¿El requisito viola algunas restricciones del dominio?
- ¿Es posible seguir el requisito hasta los objetivos del sistema o producto?
- ¿Se ha creado un índice para la especificación?

Estas son algunas de las preguntas que pueden ser útiles, no obstante, las verificaciones que han de realizarse para el conjunto de requisitos y para cada uno de ellos por individual incluye más comprobaciones, ya que como se indicó al principio de este documento, la especificación de requisitos es una parte muy importante y esencial para que el proyecto se realice con éxito.

3.2.7 Administración de los requisitos

Como ya se sabe, debido a que se ha comentado en apartados anteriores, los requisitos son cambiantes a lo largo de toda la vida del sistema, por lo tanto la fase de administración de los requisitos se encarga junto con el equipo de proyecto de identificar, controlar y dar seguimiento al conjunto de requisitos y a sus posibles modificaciones en cualquier momento del desarrollo del proyecto.

Capítulo 4: Requisitos

4.1 Definición de requisitos

La primera pregunta que se puede plantear de forma general, es la siguiente: ¿Qué se entiende cuando se habla de requisitos? Pues bien, el conjunto de requisitos de un sistema es una explicación en detalle de lo que debe ser implementado en función de las necesidades del cliente, incluyendo la forma de actuación de este. Estos son definidos en las primeras etapas de desarrollo y en determinadas ocasiones pueden condicionar el desarrollo del sistema.

A continuación se mencionan diferentes definiciones de requisito:

- En lenguaje familiar, un requisito es algo que identifica una función, restricción u otra propiedad necesaria que debe realizarse o satisfacerse para cumplir las necesidades propuestas por los usuarios de un sistema informático. [Cue02]
- Una condición o capacidad que debe cumplir o poseer un sistema o un componente del mismo para satisfacer un contrato, un estándar, una especificación, u otro documento impuesto de manera formal.

La gestión de requisitos, se trata de una actividad, la cual se inicia una vez que se observa una necesidad y dicha necesidad solo puede ser resuelta por medio de un sistema informático.

Para que un proyecto software sea exitoso, se debe entender y comprender cuál es el problema del que partimos, y cuáles son las soluciones para poder resolverlo. La forma de entender y comprender el problema, dependerá en cierto modo de la información suministrada por la persona que tenga dicho problema. En la información suministrada por el propietario del problema, deberán especificarse algunos aspectos clave como si se trata de un producto nuevo, una actualización de un producto ya existente o bien una incorporación de una nueva funcionalidad a un producto ya en uso. Hay que tener en cuenta que tanto la persona que aporta la información, como la persona que la recibe, han de ponerse de acuerdo en las características del producto o funcionalidad a desarrollar, ya que de no ser así, el resultado obtenido podría no ser el deseado.

4.2 Actividades que tienen relación con los requisitos

El conjunto de actividades de mayor importancia y relacionadas con los requisitos son “análisis y especificación de requisitos” y “gestión de requisitos”. Probablemente, nos preguntaremos ¿Cuál es la diferencia entre ambas actividades? La principal diferencia entre ambas, radica en el periodo en el que se llevan a cabo, es decir, el análisis y especificación de requisitos se realiza en el principio del proyecto, mientras que la actividad de gestión de requisitos se llevará a cabo durante toda la vida del proyecto, ya que esta engloba todo lo relacionado con el tratamiento y control de las actualizaciones. Adicionalmente a estas dos actividades, tenemos la “trazabilidad” o “facilidad de traza”, la cual está relacionada con la capacidad de establecer una relación entre dos o más productos del ciclo de vida. Es por ello, que por una parte nos aseguramos que el resultado final satisface cada uno de los requisitos del cliente, y por otra parte nos ayuda a evaluar los efectos de las peticiones de cambio de los requisitos.

Como se ha mencionado anteriormente, el paso anterior a la gestión de requisitos, es el análisis y especificación de requisitos, o lo que es lo mismo, la definición de requisitos. El documento de especificación de estos, en el desarrollo de un proyecto software, es el primer documento específico con el que trabajan los ingenieros de sistemas o las personas encargadas de contactar con el cliente final, y en el cual se engloban las necesidades del cliente.

4.3 Información acerca de las necesidades del cliente

El primer paso que se debe dar por parte del cliente/usuario del sistema a desarrollar, consiste en describir minuciosamente y en detalle las funciones que desea que tenga dicho sistema. Esta información, será recogida en un documento denominado “Declaración de necesidades del cliente”. La calidad de este documento o de este conjunto de información dependerá de los conocimientos del autor, ya que si este posee experiencia en estos tipos de trabajos, los términos utilizados serán comprensibles para el personal de informática.

En este documento, se deberán diferenciar dos partes principales: *requisitos funcionales*, que a grandes rasgos son aquellos que describen las funciones que el sistema debe realizar, y por otra parte, los *requisitos no funcionales*, que engloban las restricciones, prestaciones y limitaciones del sistema.

Normalmente, en el documento de declaración de necesidades del cliente, ambos tipos de requisitos vienen mezclados, luego es por ello, que una de las funciones que los analistas de sistemas deben realizar, consiste en diferenciar cada uno de estos.

En algunos casos, este documento puede contener ciertas carencias debidas a diferentes razones, y algunas de ellas se mencionan a continuación:

- Incluir pautas de cómo debe de realizarse la implantación o el diseño.
- Descripción de aspectos y temas que puedan tener diversas interpretaciones (ambigüedad).
- Incluir información redundante o proporcionar más cantidad de información de lo debido.
- Cometer el error de dar por supuesta alguna característica.

Es recomendable que todo este conjunto de carencias, sean detectadas lo antes posible, para que el impacto producido en los recursos sea menor. Recordamos que el coste que implica arreglar un defecto una vez realizada la entrega del producto es extremadamente alto si lo comparamos con lo que hubiera costado si este hubiera sido detectado en fases tempranas.

4.4 Documento especificación de requisitos software

Dicho documento será desarrollado a partir del documento “declaración de necesidades del cliente”, aportado por el cliente/usuario.

Probablemente, nos realicemos la siguiente pregunta: ¿Qué es una especificación? La respuesta es sencilla, una especificación es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características del sistema a desarrollar o de cualquiera de sus componentes. [Cue02]

4.5 Gestión de requisitos

La principal función de la actividad de gestión de requisitos, consiste en realizar un control de cada uno de los cambios que vayan a ser realizados en el sistema. Esto es debido, a que una vez estudiados los documentos de “Declaración de necesidades del cliente” y “Especificación de requisitos del sistema”, siempre se baraja la opción de posibles cambios, junto con la idea de que dichos documentos no son estáticos, sino dinámicos. Por lo tanto, la gestión de requisitos será la encargada del control de actualizaciones y corrección de errores entre otras.

Debido al mundo que nos rodea, estamos en un continuo cambio, es por ello que el conjunto de requisitos definidos en un principio, nunca serán firmes, sino que estos cambiarán, no solo durante la fase de desarrollo del producto, sino también una vez que este haya sido entregado, por lo tanto, la realización de un cambio se puede dar en cualquier instante del ciclo de vida del proyecto.

Las situaciones en las que nos podemos encontrar durante el tratamiento y control de cambios, son diversas, es por ello que a continuación se describen algunas de ellas:

- ❖ *Caso 1:* Sistema antiguo, en uso, del cual nunca se ha tenido un control y donde la organización quiere incluirlo -> ¿Cómo se debería actuar? -> Se tomaría el producto antiguo y el conjunto de documentación existente, como un único requisito, y desde ese punto de partida, se incorporarían los nuevos requisitos o cambios a incluir. Estas situaciones suelen ser difíciles de tratar, y normalmente conllevan grandes fracasos.
- ❖ *Caso 2:* Sistema antiguo, en uso, del cual se tiene una gestión del control desde su comienzo y donde se tiene una documentación de este completa y realizada. -> ¿Cómo se debería actuar en este caso? -> En principio la incorporación de nuevos requisitos o la realización de cambios a lo ya existente, no debería de crear ningún tipo de problemas, debido al control existente desde el comienzo y a la disciplina establecida.
- ❖ *Caso 3:* Sistema nuevo y requisitos nuevos -> ¿De qué manera se debe actuar? -> La inserción de un nuevo requisito a un sistema nuevo no conlleva ningún problema, es más, al ser tratado desde sus inicios, esto proporciona muchas ventajas.
- ❖ *Caso 4:* Sistema nuevo o antiguo, que no dispone de ningún control sobre los requisitos

4.5.1 Prácticas y principios relacionados con la gestión de requisitos

Las principales actividades que engloban la actividad de la gestión de requisitos, son las mostradas en el esquema siguiente:

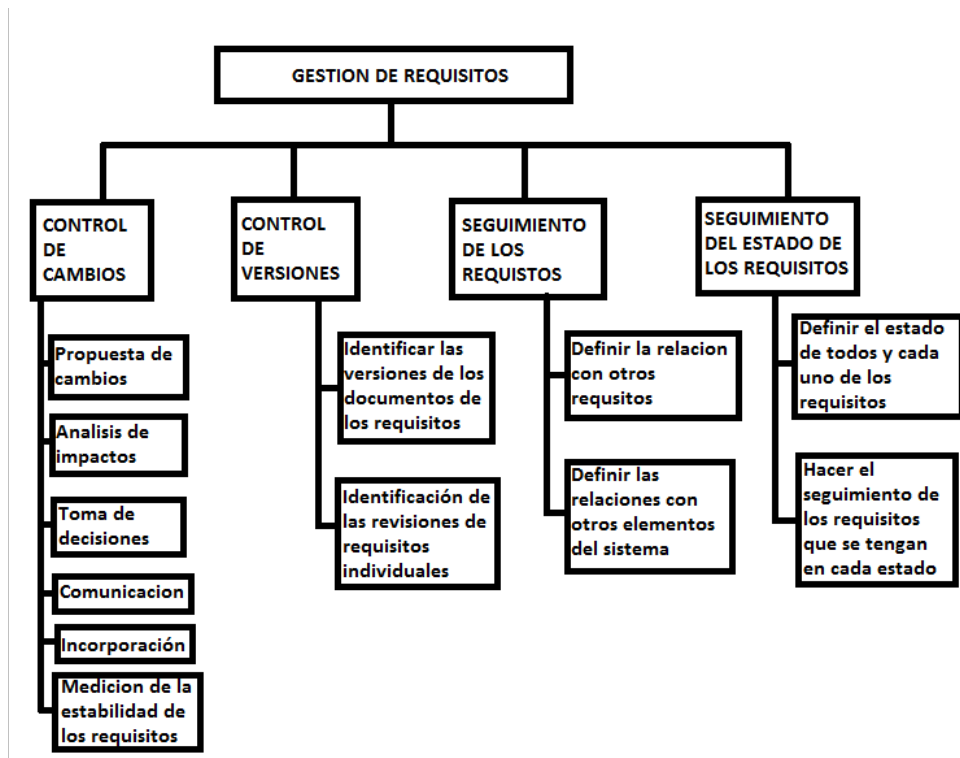


Ilustración 6: Actividades que forman la gestión de requisitos según [Cue02]

Cabe destacar, que de todas las actividades mencionadas anteriormente, aquellas que son más importantes y específicas de la gestión de requisitos son: El control de cambios, el seguimiento de los requisitos y el análisis del impacto de los cambios a los mismos.

4.5.2 Procedimientos asociados a la gestión de requisitos

Cada organización se debería encargar de especificar las actividades que engloba el procedimiento de gestión de requisitos, así como también la forma de gestionar dichas actividades dentro de la organización. Normalmente, la definición de estas actividades dependerá de dos factores, como son el tamaño de la organización y la complejidad del sistema a desarrollar.

El procedimiento de la gestión de requisitos debería contener la siguiente información, toda esta recogida en un solo documento:

- ❖ Conjunto de reglas y formas de tratar nuevos requisitos de productos ya existentes, y requisitos de nuevos productos.
- ❖ Herramientas, reglas y técnicas presentes en la organización para el control de la versión de cada uno de los requisitos y de la documentación.
- ❖ Puesta de Consideración de requisitos en la línea base.
- ❖ Reglas y formas de gestionar los cambios de productos existentes y origen de los cambios.
- ❖ Reglas o pautas a seguir para analizar el impacto de los cambios.
- ❖ Situación del estado de cada requisito, así como también persona autorizada para su modificación.
- ❖ Reglas y procedimientos de seguimiento e información del estado de los requisitos.

- ❖ Reglas y procedimientos para reflejar en los planes y compromisos del proyecto y en qué condiciones, los impactos producidos por los cambios.

4.6 Control de versiones en el conjunto de requisitos

Es imprescindible poseer información correcta acerca de la última versión actualizada o de la última versión vigente de los requisitos, ya que de no ser así, se podría trabajar con versiones no actualizadas, lo que conllevaría una pérdida de tiempo al no ser la correcta. Es de vital importancia el control de versión de un requisito para la correcta gestión de este:

- ✓ Los miembros del equipo de desarrollo deberían tener acceso a la versión vigente de los requisitos, así como a las diferentes versiones antiguas existentes.
- ✓ Los cambios realizados, además de ser comunicados a todos los afectados y claramente documentados, deberían reflejar las versiones de los requisitos afectados.
- ✓ Las versiones de los documentos de los requisitos, deberían estar puestas en la línea base así como también identificadas.

Cada una de las diferentes versiones de los documentos debe tener reflejado lo siguiente:

- La fecha en la que se realizó cada modificación.
- La persona que realizó la modificación, así como también el motivo de esta.
- Historial del conjunto de revisiones, detallando las modificaciones realizadas.

A la hora de realizar un cambio o modificación en un requisito, ha de tenerse en cuenta, que dicha modificación, de alguna manera debe ser visible a los lectores. Para ello, lo ideal sería que existieran un conjunto de normas estandarizadas dentro de la organización, con el fin de que todos los usuarios comprendieran el significado de estas.

Como comentario adicional, y en el caso particular de mi empresa, cada vez que es realizada una modificación en cualquiera de los documentos existentes, se utiliza el “control de cambios”. El control de cambios es una funcionalidad que se encuentra disponible en el paquete de Office, y en particular en Microsoft Word, donde se puede copiar, insertar, eliminar o mover texto de un documento de partida así como también insertar objetos, o cambiar el formato entre otros. Además, se ha de tener en cuenta, que el control de cambios registra el nombre del autor de cada uno de los cambios realizados, por lo que el destinatario puede observar cada uno de ellos, así como tomar la decisión de aceptarlos o rechazarlos.

Otro punto de carácter importante en el control de versiones, es la utilización del número de versión como complemento al tipo de modificación. Normalmente, la forma de utilización del número de versión, dependerá de la organización en cuestión o de alguna norma pública existente. Para la numeración, a continuación se detallan algunas propuestas a poder utilizar:

- Utilización de la *fecha de modificación* o *fecha de revisión* del documento.

La utilización de esta forma para la numeración de versiones, puede llevar a error, debido a que se puede dar el caso de que un documento sea revisado o modificado varias veces en un mismo día.

- Mediante *identificadores numéricos* de las versiones.

Capítulo 5: Documento especificación de requisitos software

5.1 ¿Qué es? ¿Por quién es utilizado?

El documento de especificación de requisitos software se empleará para notificar cada uno de los diferentes requisitos que el sistema debetener, es decir, engloba el conjunto de propiedades que el sistema debe poseer, así como también el conjunto de restricciones que este debe cumplir. Si bien es cierto, dicho documento debe estar orientado en qué debe de hacer el sistema, pero no como debe de hacerlo, ya que en este caso, ya se estaría hablando de diseño.

Se trata de una información de carácter oficial dirigida a usuarios finales, desarrolladores de software, clientes y usuarios del sistema entre otros.

Dependiendo del tipo de organización, dicho documento podrá ser nombrado de diferentes maneras como por ejemplo “Documento de especificación de requisitos (ERS)”, “especificación funcional” o “especificación de calidad”. Además es de gran importancia que la especificación de requisitos sea de gran calidad, contenga una especificación clara y coherente del sistema a realizar y sea comprensible por los usuarios con el fin de que las posibles modificaciones que tengan que ser realizadas a futuro, sean realizadas con el menor número de recursos económicos.

La especificación de requisitos, puede ser usada como referencia por una gran cantidad de individuos, entre los que se pueden destacar los siguientes:

❖ Ingenieros

- Ingenieros de Pruebas: encargados de realizar una batería de pruebas con el fin de confirmar que el sistema cumple con el conjunto de requisitos especificados en el documento.
- Ingenieros responsables del diseño e implementación: utilizan el documento para comprender el sistema que hay que desarrollar.

❖ Clientes del sistema

Utilizan o leen el documento de especificación de requisitos para verificar que este contiene el conjunto de necesidades que estos definieron en un primer momento. También con frecuencia lo suelen utilizar cuando se realiza algún tipo de modificación en el sistema bien cuando este está en desarrollo o cuando ya ha entrado en funcionamiento.

❖ Gerentes

Responsables de la planificación, cálculo de costos y la programación para el correcto desarrollo del sistema.

❖ Personal de mantenimiento

Una vez en funcionamiento el sistema, y como su nombre indica, este conjunto de personas son las encargadas de modificar y mantener este. Utilizan el documento de especificación de requisitos para entender las diferentes peculiaridades del sistema así como también la conexión entre las diferentes partes del sistema.

Igual que se ha mencionado anteriormente que dependiendo de la organización el documento de requisitos es llamado de diferente forma, también influirá en la forma de elaborar este, ya que en determinados casos este contiene una especificación del sistema en detalle, mientras que en otros casos se basa en una explicación corta. Esta discrepancia manifiesta la existencia de dos tipos de requisitos claramente diferenciados:

➤ **Requisitos del sistema**

Consiste en una explicación en detalle de las instalaciones del sistema que deben ser llevadas a cabo y las restricciones o condiciones para ser estas implementadas. Dicha especificación sobre el modo de actuación del sistema debe ser íntegra, ya que puede ser la base de un contrato para el desarrollo del sistema.

➤ **Requisitos de usuario o requisitos de partes interesadas**

Se refiere al conjunto de requisitos que detallan las características que el sistema debe tener por una parte para poderlo utilizar y por otra para poderlo integrar con sus procesos de negocio.

5.2 Documento de especificación de requisitos software con estructura estándar

Los documentos deben tener una estructura común que debe ser definida como un estándar de la compañía y debe realizarse como parte del proceso de aseguramiento de la calidad del documento. Una estructura de documento estándar debe encapsular lo que la entidad cree que es la mejor manera de organizar un documento de requisitos. Porque por la diversidad de los diferentes tipos de sistemas que puedan desarrollarse, puede que tenga que tener variantes de un único estándar en función del sistema operativo tipo y el número de lectores que se esperan.

5.2.1 Beneficios de utilizar estructura estándar

Cuando se menciona “formato estándar” para el documento de especificación de requisitos, se refiere a que los diferentes lectores de dicho documento, pueden emplear sus habilidades adquiridas de la lectura de los documentos anteriores para leer el nuevo documento de requisitos. Esto significa que pueden descubrir la información de forma sencilla así como también entender las relaciones entre las diferentes partes del documento.

Por otra parte, el uso de “formato estándar” supone una ventaja para la persona encargada de la creación del documento, debido a que de esta manera, dispone de una lista de secciones fijas a rellenar por lo que la probabilidad de olvido de escritura de información es menor. Además, los revisores o persona encargada de inspeccionar el documento pueden hacer uso del estándar con el fin de verificar si alguna sección ha quedado sin completar y de esta manera conducir el proceso para que sea revisado.

5.2.2 Mejor manera de implementar el documento con estructura estándar

La mejor estructura para el documento de especificación de requisitos, en cierto modo dependerá del tipo de organización y del tipo de sistema a desarrollar, no obstante para el inicio de

desarrollo de una estructura, es de gran ayuda el empleo de un estándar ya publicado como punto de partida.

Existe gran número de compañías que poseen su propio estándar para el documento de especificación de requisitos, pero posiblemente la norma ANSI/IEEE 830-1993 sea la más sencilla y la cual propone la siguiente estructura para el documento de especificación de requisitos:

1. Introducción
 - 1.1 Propósito del documento de requisitos
 - 1.2 Ámbito de aplicación del producto
 - 1.3 Definición, acrónimos y abreviaturas
 - 1.4 Referencias
 - 1.5 Descripción general del resto del documento
2. Descripción general
 - 2.1 Perspectiva del producto
 - 2.2 Funciones del producto
 - 2.3 Características de los usuarios
 - 2.4 Limitaciones generales
 - 2.5 Suposiciones y dependencias
3. Los requisitos específicos que cubren los requisitos funcionales, no funcionales y de interfaz. Estos deben documentar las interfaces externas, la funcionalidad, los requisitos de rendimiento de base de datos, requisitos lógicos, restricciones de diseño, atributos del sistema y las características de calidad.

Indistintamente del tipo de organización, el documento de especificación de requisitos, debe contener toda la información que se espera en este tipo de documentos, y que puede englobar algunos de los siguientes aspectos:

- ✚ Percepción global del sistema y beneficios del desarrollo del sistema.
- ✚ Glosario de términos explicando cada uno de los vocablos utilizados.
- ✚ Concepto o explicación de cada uno de los requisitos funcionales del sistema.
- ✚ Concepto o definición de los requisitos no funcionales del sistema.
- ✚ Desarrollo y restricciones de funcionamiento del sistema.

En la mayoría de las ocasiones, el estándar del documento de especificación de requisitos, estará formado por una parte estable en la que se engloban las secciones de dicho documento que siempre deben aparecer, como es el caso por ejemplo del glosario de términos y la introducción, y una parte variable que contiene el conjunto de capítulos que no tienen que ser incluidos obligatoriamente y cuyo contenido variará en función del sistema.

En el supuesto caso de que una organización no disponga de estándar para el documento de especificación de requisitos, se aconseja como buena práctica la implementación de este.

5.2.3 Costos y problemas asociados al uso del documento con estructura estándar

Los recursos económicos que conlleva la implantación de cualquier estándar en una organización son elevados, debido a que esto supone algunos meses de trabajo con el fin de estudiar e interpretar los documentos ya existentes y sacar un estándar adaptado a ellos.

Una vez que el estándar ya ha sido definido, esto acarrea unos gastos de mantenimiento, en este caso no muy elevados, ya que este precisa de una revisión y actualización periódicamente, así como también algunos controles de calidad.

Determinadas personas, en especial los desarrolladores de software, se niegan a la utilización de normas, debido a que estos argumentan que el uso de ello disminuirá la flexibilidad, así como también se incrementarán los costos del documento de especificación de requisitos. Aunque esto es un poco dudoso, hay que destacar que probablemente un aumento de los costos en la redacción del documento, implica menores costos en la lectura de este.

5.3 Pautas de cómo utilizar el documento de especificación de requisitos

El documento de especificación de requisitos siempre debe contener una sección dirigida a los lectores, en la que se explique cómo debe de ser utilizado dicho documento. Además, debe indicar qué secciones son más apropiadas dependiendo del tipo de lector, así como contener formación técnica para comprenderlo.

Beneficios

Aunque parezca absurdo la inclusión de una sección en la que se den las pautas para utilizar el documento, esto conlleva algunos beneficios que se comentan a continuación.

- *Ahorro de tiempo y mayor comprensión de requisitos:* si se detalla la forma de leer el documento de manera eficaz, los usuarios/lectores emplearán menos tiempo en la lectura y comprensión de este, lo que implica una reducción de los costos en la lectura del documento.
- Si a los lectores se les comenta lo que precisan saber para comprender el documento de especificación de requisitos, esto les ayudará a juzgar si algunos problemas de comprensión son debidos a requisitos mal expresados o a la falta de formación técnica.

Por otra parte, el documento de especificación de requisitos debe incluir al menos la siguiente información:

- Indicar qué sectores del documento están orientados a lectores específicos, así como qué secciones del documento pueden ser excluidas por otros. Se deben proponer secciones a omitir en una primera lectura, si el objetivo es captar la idea general del sistema.
- Informar acerca de la formación técnica necesaria para comprender el documento en general.

- Indicar, en el caso de existir, las diferentes dependencias entre secciones, implicando un orden de lectura del documento. En caso contrario informar de que los diferentes sectores son independientes, por lo que no existe un orden riguroso de lectura del documento.
- Indicadores apuntando a las diferentes secciones que se pueden leer en caso de querer obtener una visión general del sistema antes de entrar en detalle.
- Información acerca de los diferentes tipos de lectores a los que está dirigido el documento así como también informar de los conocimientos específicos a tener para poder entender los requisitos del sistema. Además, comunicar las secciones más complicadas de entender.

No se tiene constancia de costos y problemas significativos que pueden ser originados por la inclusión de los aspectos descritos anteriormente. Además cabe señalar que esta sección sobre cómo utilizar el documento de especificación de requisitos, debe ser breve y no se debe gastar más de una hora en escribirla.

5.4 Inclusión de un resumen de los requisitos

Todo documento de especificación de requisitos debe incluir una sección en la cual se resume el conjunto de requisitos del sistema así como también la intención o propósito del sistema.

5.4.1 Beneficios que aporta el resumen

Se parte de la base, que toda información adicional añadida al documento de especificación de requisitos, implica una serie de beneficios o ventajas para los lectores del documento. Algunos de estos beneficios, se mencionan a continuación:

- El resumen referente al conjunto de requisitos del sistema, en ocasiones puede servir de ayuda para los lectores, en el sentido de que estos pueden hallar el requisito o conjunto de requisitos de interés para ellos.
- Crear un resumen hace que se centre la atención en las necesidades críticas y puede ayudar a establecer las prioridades de los requisitos.
- El resumen de requisitos del sistema, aporta a los lectores una visión amplia de lo que están tratando de comprender.

El resumen del conjunto de requisitos del sistema puede ser realizado de diferentes maneras, las cuales se mencionan a continuación:

- Estructura de clasificación
Escribir los requisitos en estructura de clasificación, se refiere a exponer estos en forma de tabla.
- Lista numerada
Lista numerada compuesta con los requisitos de mayor importancia.
- Vista gráfica

Se refiere a la presentación de cada uno de los requisitos de mayor importancia como un nodo de un diagrama. Este modo de vista gráfica es de gran utilidad cuando se quiere conocer la relación entre grupos de requisitos.

5.4.2 Costos y problemas asociados al uso del resumen

En principio no existen costos importantes en el caso de insertar el resumen de requisitos. El costo de implementar esto dependerá del tiempo invertido en ello, el cual será proporcional al tamaño del documento, pero en situaciones normales el tiempo dedicado a esto no debe superar más de unos días de trabajo.

En cambio sí nos podemos encontrar con dos tipos de problemas en cuanto a la implementación de los requisitos, y los cuales se mencionan a continuación:

Caso 1- Cuando nos encontramos con un número elevado de requisitos: debido a que puede existir dificultad para mostrar una vista lógica de los requisitos. Esto en principio no tiene fácil arreglo, debido a que siempre un grupo elevado de requisitos implicará un problema. Sin embargo y en parte contradictorio con lo anterior, es cierto que cuanto mayor sea el documento de requisitos, más necesario es el resumen, luego merece la pena invertir el tiempo necesario en crearlo.

Caso2- Cuando ingenieros de requisitos no consideran esta parte de resumen como parte crítica del documento de especificación de requisitos: en algunas ocasiones esta situación se da, cuando los ingenieros de requisitos, al trabajar con plazos de tiempo tan ajustados, piensan que esta sección no es de gran importancia para el documento. Como solución a esto se propone que se añada dicha sección como parte del documento estándar de especificación de requisitos y que se registre esta como parte del proceso de aseguramiento de la calidad.

5.5 Realización de un caso de negocio para el sistema

Todo documento de especificación de requisitos debe de contener una sección en la que se explique claramente por una parte cuáles serán los efectos resultantes de la implantación del nuevo sistema para la organización, así como también cuales son las necesidades de esta para que sea necesario comprar el sistema.

El disponer en el documento de especificación de requisitos, de dicha información, implica una serie de beneficios que se mencionan a continuación:

- En el supuesto caso de posibles sugerencias de modificación de requisitos, puede ser utilizado el caso de negocio para evaluar si los cambios propuestos son o no viables. Se ha de tener en cuenta, que el caso de negocio puede cambiar en caso de ser necesario.
- Por otra parte, el caso de negocio será de gran utilidad para los lectores, ya que disponiendo de esta información, estos ya tienen justificación sobre el porqué de la existencia de los requisitos. Esto será eficaz sobre todo cuando se realizan sugerencias de cambio en los requisitos

Una vez que ya se ha descrito de forma general el significado de caso de uso y los posibles beneficios que ello acarrea, pueden aparecer algunas cuestiones al respecto como son:

- *¿Esta información, debe insertarse junto con la introducción en el documento de especificación de requisitos?*

No, esta información siempre aparecerá en una sección distinta a la introducción, donde como se ha mencionado anteriormente, se encontrará la información relativa al conjunto de necesidades que han dado lugar a la necesidad del sistema. Además se encontrará un conjunto de argumentos que justifican como el sistema colaborará con los objetivos de la organización. Y también en la medida de lo posible, se deben pronosticar a futuro posibles cambios de negocio y como ellos afectarán al sistema, lo que hace que los diseñadores del sistema lo planteen resistente al cambio.

- *¿Cuáles son los costos que acarrea el introducir esta pauta?*

En principio, según análisis realizados, no existen costos significativos al insertar esta sección.

5.6 Definición de un glosario de términos

5.6.1 Definición de glosario de términos

En el documento de especificación de requisitos, siempre existirán un conjunto de términos especializados, los cuales deben de ser explicados en la sección de glosario.

La Real Academia Española, define glosario como:

- Catálogo de palabras oscuras o desusadas, con definición o explicación de cada una de ellas.
- Catálogo de palabras de una misma disciplina, de un mismo campo de estudio, etc., definido o comentado.
- Conjunto de glosas o comentarios, normalmente sobre textos de un mismo autor.

5.6.2 Beneficios que aporta el glosario a los usuarios del documento

Igual que en ocasiones anteriores, el disponer de un glosario de términos en el documento de especificación de requisitos, implica un conjunto de beneficios, los cuales se comentan a continuación:

- ❖ Dependiendo del tipo de sistema, se utilizarán un conjunto de palabras las cuales solo son inteligibles por un grupo de personas expertas en ese campo. El problema con el que nos encontramos, es que, por regla general, el documento de especificación de requisitos suele ser utilizado por un amplio abanico de lectores, luego es importante explicar el significado de este grupo de palabras en el glosario de términos para que este conjunto de personas que no son expertas en el campo en cuestión puedan comprender el significado.
- ❖ La no correcta comprensión de la terminología existente en el documento, puede provocar grandes confusiones, ya que diferentes lectores son capaces de interpretar el mismo término, de manera diferente cada uno de ellos. Luego por lo tanto el disponer de un

glosario de términos donde poder encontrar el significado exacto de cierta terminología es de gran ayuda.

- ❖ En el caso de que el documento haya sido escrito por diferentes personas, se puede dar el caso de que estas hayan utilizado las mismas palabras para ser utilizadas en diferentes caminos, luego el glosario de términos, en este caso puede ser utilizado como una guía de referencia que evita estas dificultades.

5.6.3 Costos y problemas asociados al uso del glosario

El costo que ocasiona la producción y mantenimiento de un glosario de términos normalmente suele ser elevado, pero esto es compensable con los rendimientos que genera. Si en principio se dispone de un glosario de términos inicial, el esfuerzo empleado aproximadamente en la producción de este es de dos personas-mes más el tiempo que se necesita emplear en la inspección de los términos ya existentes.

Para la producción del glosario se habla con diferentes personas de la organización con el fin de verificar que el conjunto de términos importantes están contenidos en el glosario, así como también para acordar el significado de cada uno de los términos. Una vez realizado el glosario, el mantenimiento de este implica un gerente de glosario más el esfuerzo en la revisión y análisis de posibles incorporaciones de nuevos términos al glosario, lo que conlleva unos determinados costos.

En determinadas ocasiones existen algunos problemas en la definición de terminología específica, ya que algunos miembros se muestran reacios a definir abiertamente esta. En esta coyuntura, se debe aclarar que dicha definición es necesaria para la comprensión del documento así como también para comunicarse de una forma más efectiva.

5.7 Diseño del documento para facilitar la lectura

El principal objetivo que se persigue mientras se está diseñando el documento de especificación de requisitos, es que este sea de lectura sencilla, esto es, que contenga un gran volumen de información valiosa e importante, así como ordenada y clara.

Lo mencionado anteriormente, acarrea una serie de beneficios, los cuales se describen a continuación:

- Se ha de tener en cuenta que el número de veces que se lee el documento de requisitos software es suficientemente elevado, por lo que un documento que sea fácil de leer implicará menos gasto de tiempo, debido a que se lee con mayor ligereza y menos errores.
- Cuando un documento es comprensible, por una parte, los usuarios que utilizan el documento para buscar información específica o detallada, emplearán menos tiempo y por otra parte a los colaboradores les resulta más sencillo encontrar los problemas en el documento.

La creación de documentos de lectura sencilla, conlleva una serie de costos y un conjunto de problemas. En cuanto a los costos, se ha de tener en cuenta que se debe de realizar una sesión de entrenamiento en creación de documentos inteligibles, lo cual ocupará aproximadamente entre dos

y tres horas, dependiendo de la formación básica de escritura. Además, en los costos también se engloba aparte de la creación de los documentos, el mantenimiento de estos, es decir, la modificación de los documentos cuando se ha de realizar algún tipo de modificación.

5.8 Creación de un documento de especificación de requisitos fácil de cambiar

Se ha de tener en cuenta, que el documento de requisitos es útil durante toda la vida del sistema. Es por ello que dicho documento debe ser redactado de tal modo que pueda ser modificado o rectificado por los diferentes usuarios de este. Este debe ser modificable, ya que no sería viable el estar redactando un nuevo documento cada vez que se tuviera que realizar una modificación en él.

El hecho de realizar un documento fácil de cambiar, proporciona algunos beneficios, como por ejemplo ahorro de tiempo y dinero, debido a que la realización de un documento de especificación de requisitos conlleva el empleo de mucho tiempo así como también es costoso de producir, ya que engloba la redacción, verificación, impresión y distribución de dicho documento.

Existen algunas pautas que pueden ser utilizadas con el fin de realizar un documento de requisitos con mayor facilidad de modificar. Algunas de dichas pautas son las siguientes:

- ✓ Utilización de capítulos relativamente cortos.
- ✓ Mientras que se escribe el documento, intentar suprimir las referencias a otros números de páginas, ya que estas páginas en un futuro cambiarán una vez que el documento sea modificado.
- ✓ Etiquetado de todas las tablas y figuras.

Capítulo 6: Contenido del documento de especificación de requisitos

Debido a que el proyecto abarca todo lo relacionado con la actividad de requisitos software, a continuación se explica cada una de las secciones que forman el documento de especificación de requisitos software, todas ellas adaptadas según el estándar **IEEE-STD-830-1998** [IEEE98]

El documento de especificación de requisitos tiene la siguiente estructura:

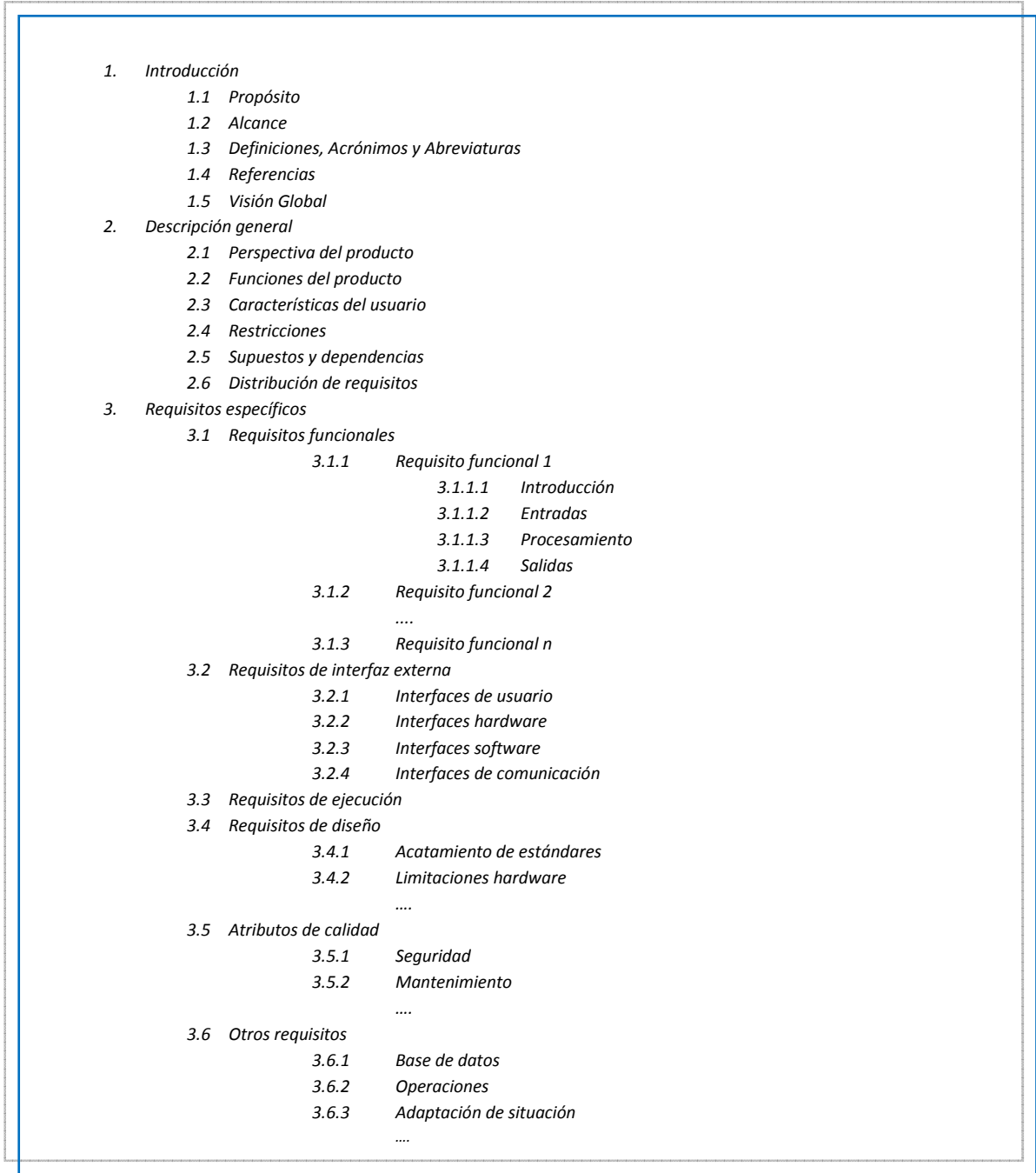
- 
- El diagrama muestra la estructura jerárquica de un documento de especificación de requisitos (ERS) según el estándar IEEE-STD-830-1998. Se trata de un árbol de lista con tres niveles principales. El primer nivel contiene tres ítems: '1. Introducción', '2. Descripción general' y '3. Requisitos específicos'. El segundo nivel expande cada uno de estos: '1.1 Propósito', '1.2 Alcance', '1.3 Definiciones, Acrónimos y Abreviaturas', '1.4 Referencias', '1.5 Visión Global' para la introducción; '2.1 Perspectiva del producto', '2.2 Funciones del producto', '2.3 Características del usuario', '2.4 Restricciones', '2.5 Supuestos y dependencias', '2.6 Distribución de requisitos' para la descripción general; y '3.1 Requisitos funcionales', '3.2 Requisitos de interfaz externa', '3.3 Requisitos de ejecución', '3.4 Requisitos de diseño', '3.5 Atributos de calidad', '3.6 Otros requisitos' para los requisitos específicos. El tercer nivel profundiza en los ítems del segundo nivel: '3.1.1 Requisito funcional 1' (con sub-ítems 3.1.1.1 Introducción, 3.1.1.2 Entradas, 3.1.1.3 Procesamiento, 3.1.1.4 Salidas), '3.1.2 Requisito funcional 2', '3.1.3 Requisito funcional n', '3.2.1 Interfaces de usuario', '3.2.2 Interfaces hardware', '3.2.3 Interfaces software', '3.2.4 Interfaces de comunicación', '3.4.1 Acatamiento de estándares', '3.4.2 Limitaciones hardware', '3.5.1 Seguridad', '3.5.2 Mantenimiento', '3.6.1 Base de datos', '3.6.2 Operaciones', '3.6.3 Adaptación de situación'. Los ítems no detallados se indican con '....'.

Ilustración 7: Estructura del documento ERS según IEEE-STD-830-1998

6.1 Introducción

Esta es la primera sección con la que nos encontramos en el documento de especificación de requisitos, y en la que se realiza una descripción global del contenido de dicho documento. Generalmente, esta sección consta de varias sub-secciones como son: propósito, ámbito, definiciones, siglas y abreviaturas, referencias y por último visión global del documento.

En este capítulo, se encontrará información relativa acerca de lo siguiente:

- ❖ Descripción relativa al comportamiento del sistema software. Dicha información será aportada principalmente mediante el uso de diferentes diagramas de estados.
- ❖ Descripción minuciosa acerca del problema que debe ser resuelto a través del software. Dicha información será proporcionada a través de estructuras de información y flujos entre otras.
- ❖ Pautas de validaciones, donde se aportan los diferentes tipos de pruebas a realizar al software, así como también los límites de rendimiento y la respuesta esperada.
- ❖ Aportación de una especificación detallada de cada función necesaria para solventar el problema.

6.1.1 Propósito

En esta sección se definirá cuál es la finalidad del documento así como también a quien va dirigido este.

Se ha de tener en cuenta, que el documento de especificación de requisitos es esencial y es el punto de arranque para cualquier sistema software. En él se reúne todo el trabajo hecho en la fase anterior, es decir, en la etapa de análisis del sistema. Debido a que es un documento que será examinado con frecuencia, este debe estar redactado de una forma sencilla de modificar.

Otro aspecto a tener en cuenta, es que este debe simplificar la tarea de verificación del cumplimiento de las especificaciones. Es por ello, que la manera más adecuada de escribir el documento sea en forma de contrato, es decir, con diferentes cláusulas organizadas y reunidas dependiendo del carácter de los requisitos.

Los requisitos específicos, se escribirán para que sean utilizados principalmente por desarrolladores y diseñadores, y se establecen a partir de los requisitos del cliente. Además estos deben ser siempre comprobables.

Debido a que el conjunto de requisitos deben de estar clasificados, una forma eficaz es estructurar estos por clases y funciones del dominio. Sabiendo que un favorable análisis de requisitos, genera grandes beneficios, a continuación vamos a estudiar las métricas existentes en la actualidad contra las que pueden ser inspeccionados los requisitos específicos.

Métricas relacionadas con el análisis de requisitos específicos

La utilización selecta de métricas, aumenta el gasto en la inspección al encaminar el proceso hacia resultados provechosos.

Por una parte, cada métrica aporta beneficios, pero por otra su recolección, el almacenamiento, y por último el análisis y estudio valen dinero y consumen tiempo, por lo que el arte de emplear métricas consiste en optimizar el costo/beneficio.

6.1.2 Alcance

El principal objetivo de esta sección consiste por una parte en describir que hará el software, así como también que es lo que no debe hacer, y por otra parte explicar para que será utilizado el software, englobando en dicha explicación aspectos como los beneficios destacables, propósitos y metas.

Además en esta subsección se incluyen otro conjunto de tareas, las cuales serán mencionadas a continuación:

- Respecto a los productos software, se reconocen el nombre de cada uno de estos que van a ser desarrollados.
- Se detalla que es lo que hacen y no hacen cada uno de los productos.
- Respecto al software, se explican las aplicaciones de este, conteniendo en dicha explicación el conjunto de metas y beneficios.

Según los tres puntos anteriores, se deduce que lo que se debe aportar en esta sección se corresponde con la descripción del producto, es por esto, que a continuación se realiza una descripción detallada sobre que debe contener la descripción del producto para que sea correcta y completa.

Descripción del producto

La descripción del producto aporta información acerca de la documentación del programa, del usuario, así como también los datos del objetivo principal de la descripción del producto.

Dicha descripción de producto, debe cumplir un conjunto de requisitos generales para que esta sea suficientemente clara, completa y sencilla con el fin de ayudar a los compradores potenciales en la valoración de la capacidad para sí mismos y del producto anteriormente a su compra. Alguno de estos requisitos generales de los que se está hablando, se describen a continuación:

1. En la descripción del producto, debe hallarse tanto el nombre como la dirección de como mínimo un proveedor.
2. En la descripción del producto debe estar identificado el producto. Dicha identificación debe contener al menos el nombre del producto y la fecha o la versión.
3. En la descripción del producto se señalará si la instalación del producto en cuestión puede ser realizada por el usuario.
4. La interfaz o productos serán reconocidos en el caso de que en la descripción de los productos se haga alusión a la interfaz con otros productos.
5. En la descripción del producto se señalará el apoyo ofrecido para el funcionamiento del producto.

6.1.3 Definiciones, acrónimos y abreviaturas

Con el objetivo de que el documento de especificación de requisitos sea comprensible, esta sección estará formada por el conjunto de definiciones, siglas y abreviaturas necesarias para alcanzar dicho propósito.

Dicha información aparecerá en forma de listado y será facilitada mediante referencias a apéndices. Se ha de prestar especial atención a la hora de aclarar términos concretos de la aplicación que estamos usando.

6.1.4 Referencias

En este apartado, nos encontramos con un conjunto de referencias, cuyos objetivos son los siguientes:

- Reconocer los documentos en función de un conjunto de campos (fecha, título y referencia).
- En cuanto a los documentos referenciados, detallar cuales son las fuentes para la obtención de estos.
- Respecto a las referencias obtenidas, detallar las fuentes de estas. Dicha información puede aportarse haciendo referencia a un apéndice u otros documentos.

Para proporcionar la información anterior, se utiliza una serie de documentación, parte de esta se resume en los siguientes puntos:

- Documentación asociada a las pruebas software.
- DDS: Descripción del diseño del software.
- PAQS: Plan de aseguramiento de la calidad del software.
- Estándar específico (IEEE 1233-1998), que se corresponde con la guía para el desarrollo de especificación de requisitos del sistema.
- Plan de documentación del usuario del software.

A continuación se muestran algunos ejemplos de modelos de referencias, aunque estos pueden variar en función de la información que se quiera recoger:

Ejemplo 1:

Referencia	Título	Fecha
[Ref.]	[Título]	[Fecha]

Tabla 8: Ejemplo de modelo de referencia

Ejemplo 2:

[IEEE93] IEEE Std 830-1993 (Revision of IEEE Std 830-1984). Software Engineering Standards Committee of the IEEE Computer Society.

The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY 10017-2394 USA.

6.1.5 Visión global

Esta sección, como su nombre indica contiene información global, esto es, por una parte nos encontramos con una especificación de lo que el documento de especificación de requisitos contiene mientras que por otra parte nos encontramos una descripción del tipo de organización de los apartados de dicho documento así como también la información contenida en ellos.

6.2 Descripción general

En este apartado nos encontramos con información, que de forma general especifica el conjunto de factores que influyen al producto. Aquí, no se fijan el conjunto de requisitos específicos, sino que se aporta una base que servirá de ayuda para secciones posteriores donde se definen el conjunto de requisitos en detalle.

Esta sección, se divide en cinco subsecciones, cada una de las cuales serán estudiadas en apartados posteriores:

- Perspectiva del producto
- Funciones del producto
- Características del usuario
- Limitaciones generales
- Supuestos y dependencias

6.2.1 Perspectiva del producto

En esta subsección del documento de especificación de requisitos se pretende describir si el producto es un producto completamente independiente, en este caso se ha de indicar que es así, o por el contrario, dicho producto es una parte de un sistema más amplio, que es lo que ocurre en la mayoría de las ocasiones, en este caso por una parte se deben reconocer las interfaces entre el software y el sistema más grande, y por otra se debe establecer una relación entre la funcionalidad del software y el conjunto de requisitos del sistema amplio.

En este apartado, además se debe encontrar una descripción acerca de la manera en la que el software opera interiormente de las diversas restricciones. Estas restricciones de las que se habla, pueden incluir por ejemplo lo siguiente:

Interfaces del sistema

En un primer lugar, se define el término “interfaz”, para que se tenga una percepción de lo que se está hablando:

Interfaz “componente de hardware o software que conecta dos o más componentes con el fin de pasar información de una a la otra.”

Bien, pues una vez comprendida la funcionalidad de una interfaz, en este apartado se deben encontrar por una parte un listado con cada interfaz del sistema, y por otra parte debe estar identificada la funcionalidad del software, con el fin de alcanzar el requisito del sistema y la explicación de la interfaz para equiparar dicho sistema.

Interfaces de usuario

Aquí se debe explicar lo siguiente:

- Todos los factores que sirven para mejorar la interfaz con el usuario que usará el sistema. Puede contener un listado con lo que hace y lo que no hace, en relación a la forma en la que se presentará el sistema a los usuarios.
- El conjunto de características lógicas de cada interfaz entre el producto y sus usuarios. Esto anterior engloba las características relativas a la configuración, necesarias para alcanzar los requisitos del software. Cuando se habla de características de configuración, se refiere por ejemplo a formatos de pantalla.

Interfaces de hardware

En este apartado se detallan el conjunto de características lógicas pertenecientes a cada interfaz entre los componentes hardware del sistema y el producto del software. Eso engloba características de la configuración, además de cubrir que dispositivos se deben soportar, como deben ser soportados estos y protocolos.

Interfaces con el software

Aquí nos debemos encontrar un detalle del empleo de otros productos necesarios (por ejemplo un determinado sistema operativo), así como también interfaces con otras aplicaciones de sistemas.

A continuación se detalla la información necesaria que debe incluir cada producto software, así como también cada interfaz:

PRODUCTO SOFTWARE	INTERFAZ
<ul style="list-style-type: none">- Nombre- Número de la especificación- Código mnemotécnico- Número de la versión- Fuente	<ul style="list-style-type: none">- La discusión de la interfaz software en relación con el producto del software.- Definición de la interfaz en relación al contenido de los mensajes y formatos.

Tabla 9: Información contenida en un producto software

Interfaces de comunicaciones

Análisis de requisitos en el desarrollo del software

En este apartado deben estar detalladas las distintas interfaces de comunicación como por ejemplo los protocolos de redes locales entre otros.

Restricciones de memoria

En esta sección nos encontraremos con información referente a la memoria primaria y a la memoria secundaria, esto es, incluirá el conjunto de características y límites aplicables en dichas memorias.

Funcionamientos

En esta sección nos tenemos que encontrar información relacionada con tipos de funcionamiento normales y especiales necesitados por el usuario, como son los siguientes:

- ✚ Especificación de los datos que son procesados por las funciones de apoyo.
- ✚ Detalle de periodos de funcionamiento que son interactivos así como también periodos de funcionamiento que son desasistidos.
- ✚ Explicación de los diferentes modos de funcionamiento dentro de la organización del usuario.
- ✚ Con respecto a la recuperación, detalle del apoyo y funcionamiento de esta.

Requisitos de adaptación al sitio

En esta sección se debe especificar lo siguiente:

- Por una parte, se debe detallar el conjunto de sitios a modificar con el fin de adecuar el software a una instalación ya existente previamente.
- Definición de los requisitos que son específicos para un sitio particular, así como también el modo operacional y la misión.

6.2.2 Funciones del producto

Esta subsección del documento de especificación de requisitos software aporta un resumen de las principales funciones que realiza el software. En ciertas ocasiones, dicho resumen que es imprescindible para esta sección puede tomarse de las especificaciones de alto nivel. Este resumen del que se está hablando, debe de cumplir lo siguiente:

- ✓ El conjunto de funciones deben estar organizadas de tal manera que cualquier persona que lea el documento por primera vez sea capaz de entenderlo a la perfección.
- ✓ Pueden ser utilizados elementos textuales o gráficos con el fin de presentar las diversas funciones y sus relaciones. Esto anterior, se refiere a una presentación de relaciones lógicas entre variables y funciones, pero no un diseño del producto.

6.2.3 Características del usuario

En esta sección del documento, nos encontramos con una descripción minuciosa acerca de las características generales que los usuarios del producto deben tener, incluyendo en esto el nivel de conocimiento y experiencia, las características de las tareas y trabajos de usuario, las características físicas del usuario y por ultimo último las características psicológicas del usuario.

A continuación se muestra una tabla aclarativa que contiene el conjunto de aspectos que se valoran en cada grupo de características anteriores:

CARACTERÍSTICAS	ASPECTOS ENGLOBADOS
Características físicas del usuario	<ul style="list-style-type: none">- Edad- Género (masculino o femenino)- Discapacidad (sí o no)
Características psicológicas del usuario	<ul style="list-style-type: none">- Grado de motivación- Aptitud mostrada hacia el trabajo- Estilo cognitivo (se refiere al modo de caracterización de recordar, pensar y percibir)
Nivel de conocimiento y experiencia	<ul style="list-style-type: none">- Conocimientos de informática- Nivel educativo (licenciado, graduado...etc.)- Experiencia con aplicaciones- Nivel de lectura- Habilidades
Características de las tareas y trabajos de usuario	<ul style="list-style-type: none">- Importancia de la tarea- Tipo de utilización de la aplicación- Frecuencia de utilización- Repetitividad de la tarea- Tasa de rotación de empleados- Capacidad prevista- Categoría del trabajo

Tabla 10: Características generales de los usuarios

La tabla anterior, es una lista de comprobación, cuyo principal propósito consiste en garantizar que el conjunto de características básicas de los usuarios son conocidas. Como regla general, el usuario con menores niveles en el conjunto de características anteriores, siempre requiere más aclaraciones así como también más ayudas.

6.2.4 Restricciones

Como su nombre indica, en este apartado del documento de especificación de requisitos software, nos encontraremos con una explicación general acerca de cualquier aspecto que pueda restringir las opciones de los diseñadores. Según el IEEE Std 830-1998[IEEE98], estas limitaciones o restricciones engloban lo siguiente:

- Políticas reguladoras
- Limitaciones del hardware
- Interfaces a otras aplicaciones
- El funcionamiento paralelo
- Funciones de auditoría
- Funciones de control
- Requisitos del lenguaje
- Requisitos de fiabilidad

- Credibilidad de la aplicación
- La seguridad y consideraciones de seguridad

Algunos ejemplos de restricciones son por ejemplo los siguientes:

- El sistema deberá de realizar una conexión a una base de datos específica.
- El ordenador donde se ejecute el sistema, debe tener instalado Java con versión 7 update 45.

6.2.5 Supuestos y dependencias

Como ya sabemos, el documento de especificación de requisitos contiene un grupo de requisitos software, bien pues en este apartado nos encontraremos un listado que engloba el conjunto de factores que incumben a dichos requisitos.

Estos factores de los que se habla, no son referentes a limitaciones en el diseño del software, sino que detallan aquellos factores, que en caso de sufrir una modificación, pueden afectar a los requisitos.

Un ejemplo para clarificar este apartado, es el siguiente: en mi compañía, el sistema donde todos los agentes de mercado se autentican con el fin de introducir sus ofertas para los diferentes mercados eléctricos, precisa que el ordenador donde se utiliza tenga instalado el sistema operativo Windows 7 así como también que la máquina virtual de Java utilizada sea la 1.7 update 45. Sabido esto, si estos detalles minuciosos son modificados en la compañía, puede ser que sea necesario comprobar y modificar los requisitos técnicos, ya que el documento de especificación de requisitos software, sería necesario que fuera modificado de acuerdo a las modificaciones realizadas.

6.2.6 Distribución de requisitos

En este apartado nos encontramos con un listado, el cual contiene el conjunto de requisitos que son guardados para versiones futuras.

6.3 Requisitos específicos

Esta sección del documento de especificación de requisitos, es una de las de mayor peso, ya que encierra el conjunto de requisitos software descritos con un nivel de detalle suficientemente grande como para conceder a los diseñadores realizar un diseño del sistema de tal forma que este satisfaga dichos requisitos. Por otra parte esta descripción en detalle de los requisitos software también permitirá a los auditores comprobar que el sistema satisface o cumple los requisitos definidos. Cada requisito especificado en esta sección debe ser externamente perceptible por los operadores, usuarios u otros sistemas externos.

Cada requisito software, debe estar descrito de tal manera que aporte la siguiente información:

- Descripción de cada una de las entradas en el sistema, refiriéndose al estímulo.
- Descripción de cada salida proporcionada por el sistema como respuesta a la entrada.
- Conjunto de funciones hechas por el sistema en el apoyo de la salida o en la salida a una entrada.

REQUERIMIENTO Nro.04		
Los usuarios deben poder acceder al sistema		
Usuario	x	
Sistema		
Funcional	x	
No funcional		
Pre-condición		Post-condición
Tener asignada la condición de agente		Acceder el sistema así como poder usar todas las funcionalidades de este.
Fecha creación	12/11/2014	Responsable
Fecha procesamiento	25/01/2015	Dalila García Notario
Modificación		Causa/ responsable

Tabla 11: Ejemplo de requisitos software

Esta sección es la que nos encontramos, como ya se ha comentado con anterioridad, es la parte más importante y la más amplia del documento de especificación de requisitos, y en ella se aplican los siguientes principios:

- ❖ Cada uno de los requisitos debe de poder ser identificado particularmente.
- ❖ Los requisitos específicos deben poseer referencias cruzadas a otros documentos más recientes que los relacionan.
- ❖ Debe prestarse especial atención a la organización de los requisitos software con el fin de aumentar en la medida de lo posible todo lo que se pueda la legibilidad.
- ❖ Los requisitos específicos deben ser declarados conforme al conjunto de características definidas en el apartado de “Características del usuario”.

6.3.1 Requisitos funcionales

En la descripción general aportada como introducción a los requisitos específicos, se ha comentado la información que debe poseer cada requisito software (entre toda ella, entradas y salidas). Bien pues en este apartado de requisitos funcionales se deben explicar las principales acciones que se deben llevar a cabo en el software, aceptando y procesando las entradas, así como también procesando y produciendo las salidas.

Por regla general, siempre el conjunto de requisitos funcionales son descritos comenzando con las palabras: “El sistema debe...”. Estos engloban lo siguiente:

- Comprobar la validez sobre las entradas.

- La relación de las salidas referentes a las entradas, conteniendo:
 - Las fórmulas aplicadas a las entradas para obtener las salidas.
 - Las secuencias de entradas/salidas.
- En situaciones no normales, las respuestas que se aportan, incluyendo:
 - Capacidad de comunicación.
 - Empleo de errores y técnicas de recuperación.
 - Desbordamiento (Overflow).
- Sucesión precisa de las operaciones

Debido a que el término “Overflow” no es utilizado con frecuencia, a continuación se aporta una breve descripción de dicho término:

Overflow: la definición de esta palabra inglesa significa “desbordamiento”. En términos generales tiene relación con un exceso de datos que corren el riesgo de ser extraviados o transferidos.

6.3.2 Requisitos de interfaz externa

Este apartado consiste en una descripción minuciosa y al detalle de cada una de las entradas y salidas del sistema software. Esta sección debe complementar las descripciones de los apartados anteriores, pero nunca debe existir información repetida.

La descripción detallada que se comenta en el párrafo anterior, debe estar estructurada de la siguiente manera:

- ✚ Nombre del elemento
- ✚ Explicación del propósito
- ✚ Fuente de entrada o destino de salida
- ✚ Tiempos
- ✚ Formato de datos
- ✚ Unidades de medidas
- ✚ Relaciones con otras entradas o salidas
- ✚ Formato de los diferentes comandos
- ✚ Formato u organización de la pantalla
- ✚ Formato u organización de las ventanas
- ✚ Tolerancia, exactitud y /o rango válido
- ✚ Fin de mensajes

6.3.3 Requisitos de ejecución

Este apartado debe especificar y diferenciar entre requisitos estáticos y requisitos dinámicos que se colocaron en el software o en la interacción humana con el software en conjunto.

En ocasiones los requisitos estáticos son reconocidos en una sección separada denominada “Capacidad” y los requisitos dinámicos cabe la posibilidad de que contengan aspectos como el número de transacciones, volumen de datos a ser procesados con un tiempo de respuesta máximo para condiciones de trabajo normales, y para condiciones de trabajo extremas, así como también estos contienen las tareas.

6.3.4 Requisitos lógicos de la base de datos

A grandes rasgos este tipo de requisitos especifican los requisitos lógicos para cualquier información que es guardada en un banco de datos. Esto anterior puede llegar a englobar lo siguiente:

- Tipos de información utilizada por diversas funciones
- Habilidades de acceso
- Frecuencia de uso
- Entidades de datos y sus relaciones
- Restricciones de integridad
- Retenciones de datos

6.3.5 Restricciones de diseño

Esta sección del documento de especificación de requisitos software detalla los requisitos relacionados con las regulaciones o estándares existentes.

Las restricciones de diseño detallan cuales son los límites para realizar el diseño de la aplicación. El fin de este conjunto de requisitos no es suplantar el proceso de diseño, únicamente especifican las condiciones exigidas por el cliente ya sea para el proyecto, entorno u otra circunstancia.

6.3.5.1 Aceptación de las normas

Como se ha comentado anteriormente, en este apartado nos encontramos con el conjunto de requisitos procedentes de regulaciones o estándares ya existentes. Dichos requisitos pueden contener lo siguiente:

- Formato del informe.
- Procedimientos de contabilidad.
- Nombres de los datos.
- Lineamientos de la auditoría.

6.3.5.2 Atributos del software del sistema

Existen un conjunto de propiedades del software que puede servir como los requisitos. Es de gran importancia que dichos requisitos se detallen con el fin de que su logro pueda comprobarse objetivamente.

Dichos atributos de los que se está hablando, se corresponden con propiedades del sistema, cuyo conjunto forman un requisito. A continuación se proporciona una lista de alguno de ellos.

6.3.5.2.1 Fiabilidad

Debe detallar los factores necesarios para determinar la fiabilidad requerida del sistema del software en el momento que se debe realizar la entrega.

Según la **norma ISO/IEC 9126:2004** se define la fiabilidad como: “Conjunto de atributos que influyen en la capacidad del software para mantener su nivel de desempeño bajo las condiciones establecidas para un periodo determinado de tiempo”.

6.3.5.2.2 Seguridad

Detallar los factores que salvaguardan el software del ingreso accidental por parte de usuarios, uso, cambios, revelación o destrucción. En todo momento, debe asegurarse que el conjunto íntegro de los datos están protegidos, es decir, están seguros. Para conseguir este fin, en determinadas ocasiones es necesario llevar a cabo algunas de las siguientes pautas:

- Uso de algunas técnicas de cifrado;
- Limitar comunicaciones entre ciertas áreas del programa;
- Asignar funciones específicas a diferentes módulos;
- Comprobar la integridad de los datos para variables críticas;
- Disponer de históricos de datos, así como también del log de entradas;

6.3.5.2.3 Usabilidad

Según la **norma ISO/IEC 9126:2004**, la usabilidad es definida como: “Conjunto de atributos que influyen en el esfuerzo necesario para su uso, y en la evaluación individual de tal uso implícita por un conjunto de usuarios”.

6.3.5.2.4 Portabilidad

Según la **norma ISO/IEC 9126:2004**, la portabilidad se define como: “Conjunto de atributos que influyen en la capacidad del software para ser transferido de un entorno a otro”.

En este apartado se deben detallar atributos o propiedades del software que vincula a la simplicidad de mantenimiento del software, englobando lo siguiente:

- Utilización de un idioma portable ya probado con anterioridad;
- Utilización de un sistema operativo especial;
- Utilización de un compilador especial o un subconjunto de lenguajes,
- Porcentaje de código cliente-servidor;
- Porcentaje de componentes con código cliente-servidor

6.3.5.2.5 Mantenimiento

En este apartado nos encontraremos con el conjunto de atributos de software que se refiere a la facilidad de conservación del software.

6.3.5.3 Organización de los requisitos específicos

La principal finalidad de la organización de requisitos específicos consiste en organizar estos de tal manera que sean comprensibles. A menudo el método utilizado para organizar los requisitos específicos está relacionado con la probable arquitectura de la aplicación.

6.3.5.3.1 Modo del sistema

Existen sistemas los cuales se comportan de maneras distintas dependiendo de los diferentes modos de operación. Ahora se mostrará la plantilla que debe ser utilizada cuando la organización depende del modo de operación. En este caso en particular, existen dos versiones posibles que pueden ser utilizadas, las cuales se muestran a continuación:

Plantilla A1:

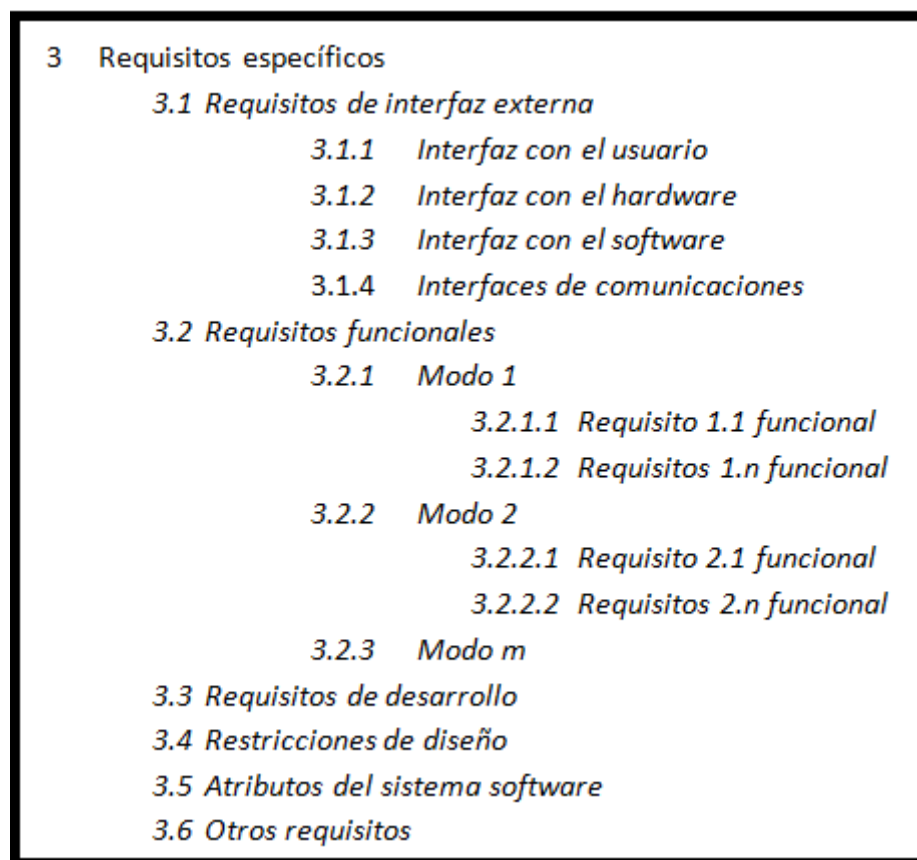


Ilustración 8: opción 1, documento ERS cuando la organización depende del modo de operación. Adaptado IEEE-STD-830-1998

Como se puede observar en el conjunto de plantillas que se van a proporcionar, el tercer apartado es el único que se describe, ya que este es el que sufre variaciones, ya que el resto es idéntico al de la plantilla general descrita.

Plantilla A2:

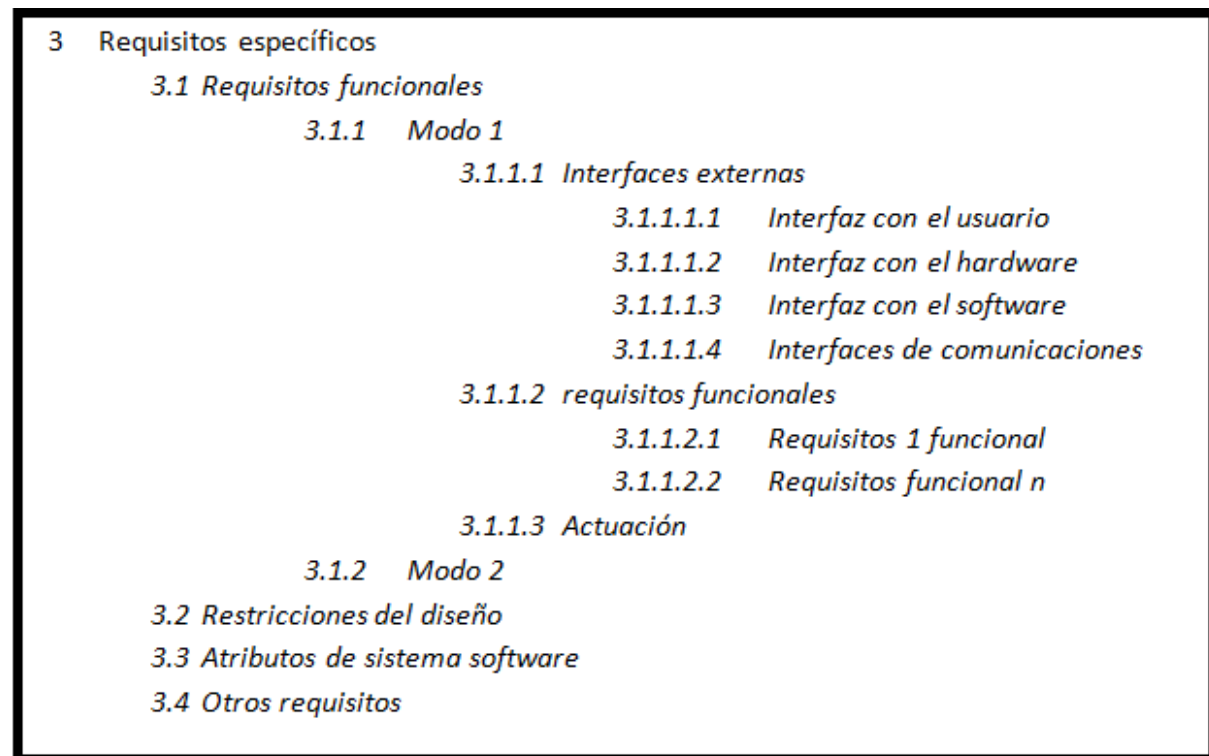


Ilustración 9: opción 2, documento ERS cuando la organización depende del modo de operación. Adaptado IEEE-STD-830-1998

6.3.5.3.2 Clases de usuario

Existen sistemas que suministran diferentes juegos de funciones a las diferentes clases de usuarios. Esto es, dependiendo de a que esté dirigido, las funciones a destacar serán unas u otras.

Ahora se mostrará la plantilla que debe ser utilizada cuando la organización depende de la clase del usuario:

Plantilla A3:

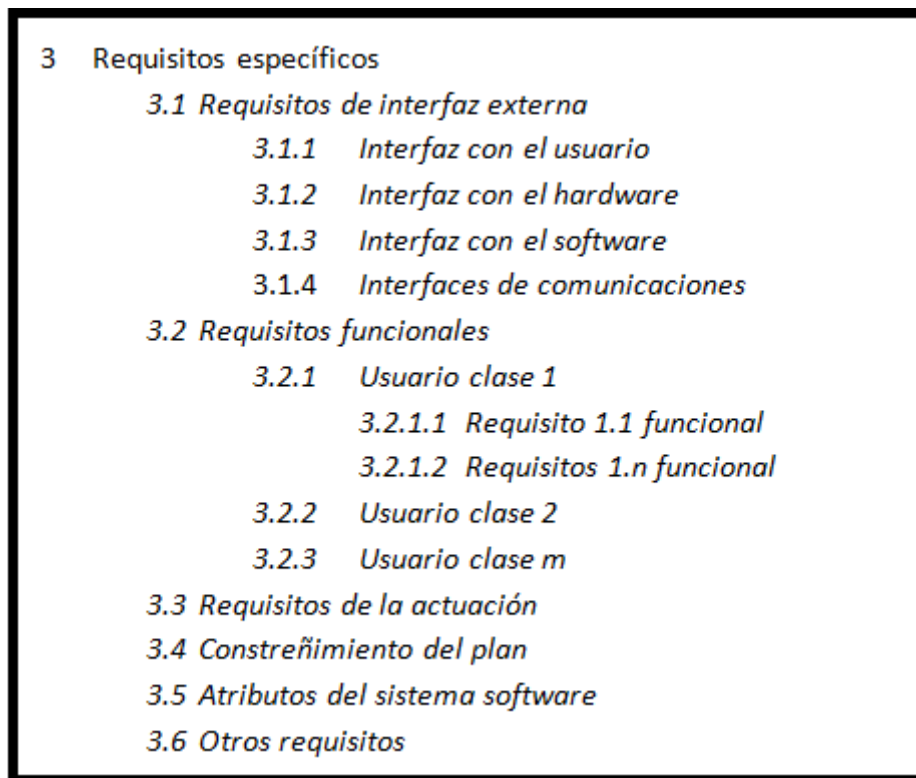


Ilustración 10: Estructura del documento ERS cuando la organización depende de la clase de usuario.
Adaptado IEEE-STD-830-1998

6.3.5.3.3 Objetos

Los objetos son sujetos del mundo real que poseen una contraparte dentro del sistema. A su vez, a cada objeto está asociado un conjunto de funciones ejecutadas por dicho objeto y un conjunto de atributos. Existen varias formas de nombrar a las funciones, como son procesos, servicios o métodos.

Por ejemplo, en mi lugar de trabajo, referente a un sistema de subasta de energía, el conjunto de objetos englobaría al conjunto de operadores del mercado, los diferentes agentes que participan en este, los ordenadores y la sala de operación entre otros.

Ahora se mostrará la plantilla que debe ser utilizada cuando la organización depende del objeto:

Plantilla A4:

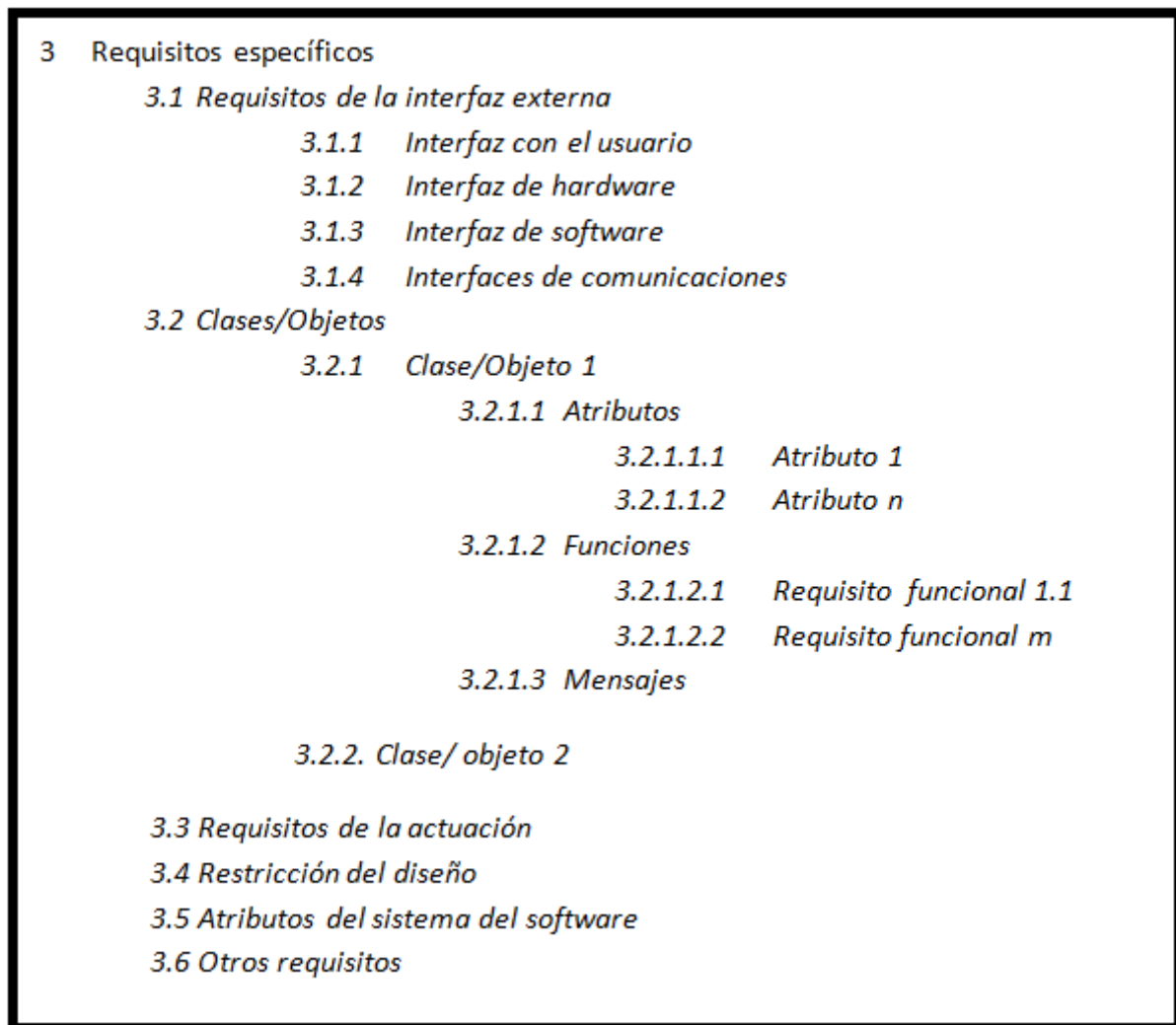


Ilustración 11: documento ERS cuando la organización depende del objeto. Adaptado IEEE-STD-830-1998

6.3.5.3.4 Rasgo

Un rasgo es un servicio requerido por el sistema que puede requerir una sucesión de entradas en un orden específico, con el fin de obtener la salida deseada.

Por ejemplo, en mi caso, en el sistema de subasta de energía, para que se produzca esta, se precisa un sistema donde insertar las ofertas (web de agentes en este caso), una oferta de compra proveniente de un Agente1 y una oferta de venta proveniente de un Agente2. Bien, pues en este caso, los rasgos incluyen lo siguiente:

1. Inserción por ejemplo del Agente1 de una oferta válida de compra.
2. Recepción y procesamiento en la web de agentes de dicha oferta.
3. Inserción del Agente2 de una oferta de venta.
4. Recepción y procesamiento en la web de agentes de una oferta válida de venta.

6.3.5.3.5 Estímulo

Determinados sistemas pueden organizarse de mejor manera detallando sus funciones en términos estímulo estímulo-respuesta.

Hablando de este tema, por ejemplo en mi empresa, el sistema está organizado de tal manera, que actúa de la siguiente forma:

- Cuando se recibe una oferta que no cumple las limitaciones máximas y mínimas declaradas en el fichero correspondiente, la oferta directamente es descartada con el fin de que no entre en el proceso de casación.

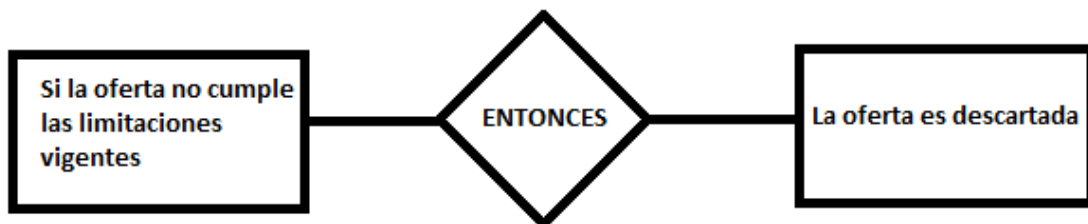


Ilustración 12: Ejemplo de diagrama cuando las limitaciones no se cumplen

- Cuando una oferta es insertada en el sistema con unas credenciales distintas al certificado instalado en el ordenador del agente que realiza la orden de inserción, la oferta es directamente descartada.

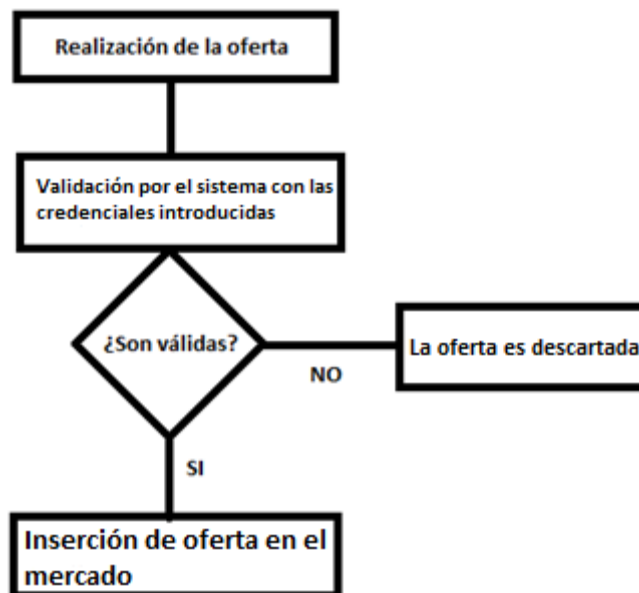


Ilustración 13: Ejemplo de diagrama que relaciona el certificado con las credenciales utilizadas

Ahora se mostrará la plantilla que debe ser utilizada cuando la organización depende del estímulo:

Plantilla A6:

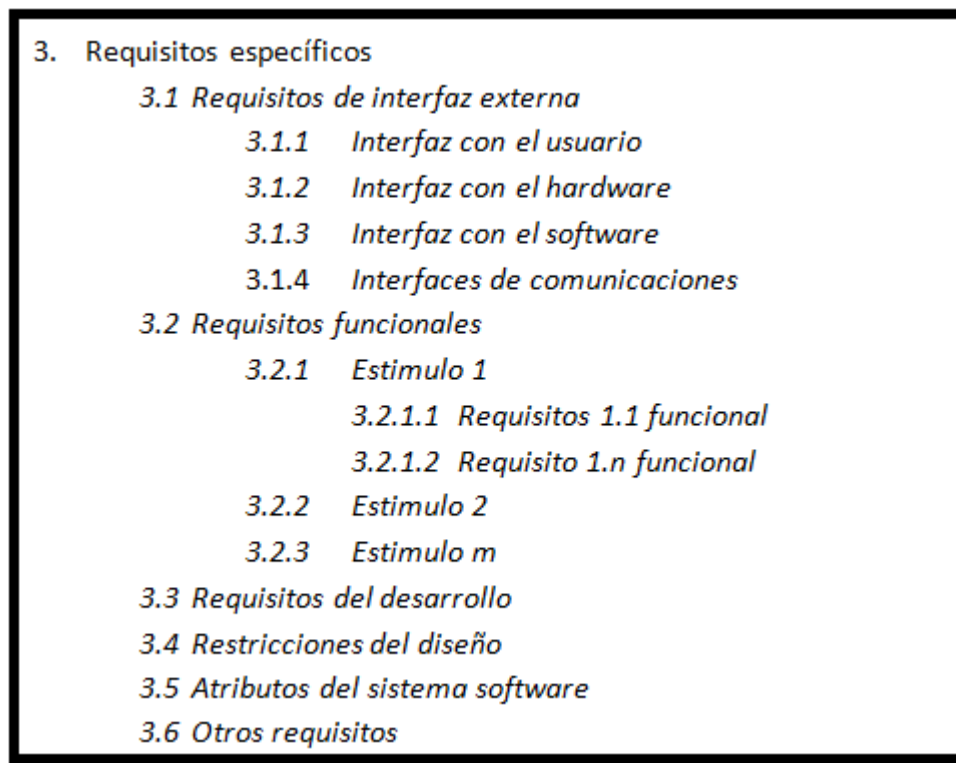


Ilustración 14: documento ERS cuando la organización depende del estímulo. Adaptado IEEE-STD-830-1998

6.3.5.3.6 Contestación

Determinados sistemas pueden estructurarse de mejor manera detallando cada una de las funciones en el apoyo de la creación de una contestación.

En esta sección se ha de tener en cuenta que la plantilla a utilizar es la misma que la usada en el apartado de estímulos (Plantilla A6).

6.3.5.3.7 Jerarquía funcional

Cuando ninguno de los esquemas anteriores explicados son de gran utilidad, la funcionalidad global puede organizarse en una jerarquía de funciones organizada con rendimientos y entradas comunes.

Normalmente se utilizan diagramas de flujos y diccionarios de datos con el fin de poder visualizar las relaciones existentes entre las funciones y datos.

Ahora se mostrará la plantilla que debe ser utilizada cuando la organización depende de la jerarquía funcional:

Plantilla A7:

- 3. Requisitos específicos
 - 3.1 Requisitos de interfaz externa
 - 3.1.1 Interfaz con el usuario
 - 3.1.2 Interfaz con el hardware
 - 3.1.3 Interfaz con el software
 - 3.1.4 Interfaces de comunicaciones
 - 3.2 Requisitos funcionales
 - 3.2.1 Flujo de la información
 - 3.2.1.1 Diagrama de flujo de datos 1
 - 3.2.1.1.1 Entidades de los datos
 - 3.2.1.1.2 Procesos pertinentes
 - 3.2.1.1.3 Topología
 - 3.2.1.2 Diagrama de flujo de datos n
 - 3.2.1.2.1 Entidades de los datos
 - 3.2.1.2.2 Procesos pertinentes
 - 3.2.1.2.3 topología
 - 3.2.2 Descripciones del proceso
 - 3.2.2.1 Proceso 1
 - 3.2.2.1.1 Entidades de datos de entrada
 - 3.2.2.1.2 Algoritmo o formula del proceso
 - 3.2.2.1.3 Entidades de los datos afectados
 - 3.2.3 Construcción de las especificaciones de los datos
 - 3.2.3.1 Estructura 1
 - 3.2.3.1.1 Tipo de registro
 - 3.2.3.1.2 Elector presenta
 - 3.2.4 Diccionario de datos
 - 3.2.4.1 Datos elemento 1
 - 3.2.4.1.1 Nombre
 - 3.2.4.1.2 Representación
 - 3.2.4.1.3 Formato
 - 3.2.4.1.4 Precisión
 - 3.2.4.1.5 Rango
 - 3.3 Requisitos del desarrollo
 - 3.4 Restricciones del diseño
 - 3.5 Atributos del sistema software
 - 3.6 Otros requisitos

Ilustración 15: documento ERS cuando la organización depende de la jerarquía funcional. Adaptado IEEE-STD-830-1998

6.3.5.4 Comentarios adicionales

El conjunto de técnicas organizacionales aportadas en el punto anterior pueden ser adecuadas siempre y cuando un nuevo documento de especificación de requisitos software se considere. En dicho caso, según las necesidades específicas del sistema, se han de organizar los requisitos específicos para jerarquías múltiples.

Existen diversos métodos y herramientas de ayuda automatizadas con el fin de colaborar en la documentación de requisitos.

6.3.6 Información de apoyo

La información de apoyo contenida en el documento de especificación de requisitos ayuda para que este sea utilizado. En este apartado se engloban las tablas de contenidos, apéndices e índices.

6.3.6.1 Tabla de contenidos e índice

Como su nombre indica, una tabla o índice te concede organizar y agrupar un determinado trabajo por diferentes temas o secciones. A los lectores les beneficia debido a que a estos les simplifica la localización de un tema de interés.

6.3.6.2 Apéndices

Como ya sabemos, un apéndice es un conjunto de información que se añade al final de un documento con el fin de completar algunos de los apartados incluidos en dicho documento. Normalmente se suele tratar de información no esencial.

En muchas ocasiones, los apéndices no son imprescindibles, no obstante, estos pueden englobar lo siguiente:

- Aportación de información con un alto grado de especificación que a menudo ayuda a los lectores del documento de especificación de requisitos.
- Definición de los problemas a ser solucionados por el software.
- Instrucciones de empaquetamiento especial para el código y los medios de comunicación para reunir la seguridad, exportar la carga inicial u otros requisitos.
- Ejemplos de formatos de entrada/salida, descripciones del análisis del costo que fueron estudiados o resultados de estudios del usuario.

Capítulo 7: Obtención de requisitos

Una de las principales fases del proceso de ingeniería de requisitos es la obtención de requisitos, de forma general, es la fase donde usuarios finales, ingenieros software y clientes trabajan en conjunto con el fin de descubrir el dominio de la aplicación y las restricciones hardware entre otras, a groso modo en esta fase el principal objetivo consiste en indagar acerca de lo que los usuarios necesitan. En esta fase no solo basta con poseer información acerca de las necesidades del cliente, sino que se requiere de un análisis minucioso acerca de la organización, del dominio de aplicación, así como también de cómo será utilizado el sistema.

Este proceso implicará a un gran número de personas directa o indirectamente dentro de la organización. Como se ha mencionado en apartados anteriores, cuando se hable de stakeholders, nos referiremos a cualquier persona o grupo de personas que se vea afectado por el sistema.

7.1 Problemas asociados a la obtención e interpretación de los requisitos

Conseguir e interpretar los requisitos del conjunto de personas afectadas por el sistema es complicado por diferentes razones, y algunas de ellas se mencionan a continuación:

- ❖ Normalmente el conjunto de personas encargadas de ellos, expresan los requisitos de forma natural y con sus propios términos, por lo que los ingenieros de requisitos, sin ningún conocimiento previo del negocio del cliente, deben entender y comprender dichos requisitos.
- ❖ Normalmente el grupo de personas afectadas por el sistema raras veces conocen lo que el sistema informático debe realizar o lo que se quiere obtener de este, es por ello, que en la mayoría de las ocasiones les es embarazoso indicar que es lo que desean que haga el sistema incluso en ocasiones llegan a realizar peticiones irreales debido a que no se percatan del coste de dichas demandas.
- ❖ Debido a la diversidad de personas afectadas por el sistema, cada uno de los diferentes stakeholders tienen diferentes requisitos, todos ellos manifestados de diversas maneras. Es por esto, que los ingenieros de requisitos deben tener en cuenta todos ellos provenientes de las diversas fuentes así como descubrir la relación y conflictos entre ellos.

7.2 Modelo general del proceso de obtención y análisis de requisitos

A continuación se muestra un modelo general del proceso de obtención y análisis de requisitos. Como se puede observar en la ilustración 16, este modelo se basa en cuatro fases claramente diferenciadas: descubrimiento de requisitos, clasificación y organización de requisitos, ordenación por prioridades y negociación de requisitos y por último documentación de requisitos. No obstante, cada entidad tendrá su propia versión del modelo del proceso de obtención y análisis de requisitos, ya que esta dependerá de diferentes factores como tipo de sistema a desarrollar y estándares utilizados entre otros.

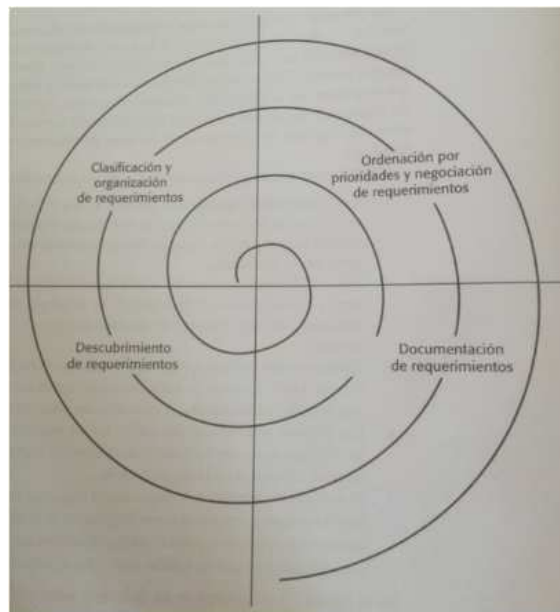


Ilustración 16: Proceso de obtención y análisis de requisitos tomado de [Som05]

Como se puede observar en la figura anterior, dicho modelo de proceso es iterativo con retroalimentación continua de cada actividad, en el que el analista comprende mejor los requisitos en cada vuelta de la espiral.

A continuación se describen cada una de las fases o actividades del proceso de forma más extendida:

- **Descubrimiento de requisitos:** de forma general, en esta fase se obtienen el conjunto de requisitos a través de las personas afectadas por el sistema o stakeholders. La documentación y requisitos del dominio de los stakeholders también se descubren en esta fase.
- **Clasificación y organización de requisitos:** en esta actividad, se toma el conjunto de requisitos obtenidos en la fase anterior, todos ellos sin ninguna estructura, y se ordenan en grupos lógicos.
- **Ordenación por prioridades y negociación de requisitos:** siempre y cuando el grupo de stakeholders sea grande, se dará la circunstancia de que varios requisitos entren en conflicto, es por ello que en esa fase, como su nombre indica, se ordenarán los requisitos por prioridad así como también se descubrirán y se solucionarán los requisitos en conflicto mediante pacto o negociación. Está claro que cada uno de los stakeholders tendrá distintas consideraciones acerca de los requisitos, pero durante el proceso se deben de realizar negociaciones con ellos, con el fin de llegar a un pacto.
- **Documentación de requisitos:** todo el conjunto de requisitos anteriores se documenta, creando así el documento de requisitos software, y una vez hecho esto se entra de nuevo en la siguiente vuelta de la espiral.

De forma particular, la fase de obtención de requisitos consiste en recopilar toda la información posible acerca del nuevo sistema a desarrollar, así como también información sobre el

sistema actual con el fin de obtener de toda esta información, los requisitos de usuario y los requisitos de sistema. Las fuentes de información utilizadas serán diversas, desde especificaciones de sistemas similares ya existentes hasta la información proveniente de los stakeholders. En cuanto a estos últimos, la idea consiste en relacionarse con ellos mediante la observación, entrevistas y con la posibilidad de utilizar prototipos, métodos de análisis estructurado y escenarios entre otros con el fin de averiguar requisitos.

Resumiendo, el conjunto de requisitos entonces será obtenido del dominio de la aplicación, de los stakeholders y del conjunto de sistemas que interactúan con el sistema a especificar. Estas fuentes de requisitos se pueden representar como puntos de vista del sistema, donde en cada una de ellas se expone un grupo de requisitos para el sistema, indicando que cada grupo de estos no son independientes entre sí, sino que normalmente suelen coincidir y por ello tienen requisitos en común.

7.3 Técnicas para realizar la obtención de requisitos

En cuanto a las posibles técnicas a utilizar, existen un conjunto de técnicas tradicionales conocidas, que engloban las entrevistas, puntos de vista, escenarios, introspección, grupos focales, casos de uso, etnografía y análisis de protocolos. En las líneas posteriores se explica en detalle en qué consiste cada una de ellas.

Puntos de vista

La principal ventaja de la utilización de puntos de vista es que pueden ser detectados conflictos entre requisitos, debido a que se poseen requisitos sugeridos de diferentes fuentes o stakeholders. En general, los puntos de vista, pueden ser utilizados para organizar las diferentes fuentes de información y los stakeholders.

Existen tres tipos comunes de puntos de vista, y los cuales se describen a continuación:

- ❖ *Puntos de vista indirectos*: lo constituye el conjunto de stakeholders que no utilizan el sistema ellos mismos, pero que de algún modo influyen secundariamente en los requisitos. Este punto de vista aportará requisitos y restricciones organizacionales de alto nivel.
- ❖ *Puntos de vista del dominio*: representan restricciones y características del dominio que afectan a los requisitos del sistema. En este caso se obtendrán restricciones del dominio que se aplican al sistema.
- ❖ *Puntos de vista de los interactuadores*: representan al conjunto de sistemas o personas que interactúan directamente con el sistema. Estos aportan requisitos detallados que cubren las características del mismo. Los casos de uso y escenarios son técnicas eficaces para obtener requisitos para este tipo de puntos de vista.

En la mayoría de las ocasiones, el reconocimiento del conjunto de puntos de vista significativos, es complicado, es por ello que para que este proceso sea más sencillo, se debería tratar de encontrar puntos de vista más específicos de los tipos siguientes, estos traducidos y adaptados de [Som05]:

- Receptores y proveedores de servicios al sistema.
- Debe existir una interacción entre el sistema y el sistema a especificar.

- Las regulaciones y estándares que se emplean al sistema.
- Las fuentes de los requisitos no funcionales y de negocio del sistema.
- Los puntos de vista de la ingeniería que manifiestan los requisitos de las personas que tienen que desarrollar, administrar y mantener el sistema.
- Los puntos de vista del marketing y otros que generan requisitos sobre las características del producto esperadas por los clientes y como cómo el sistema debería reflejar la imagen externa de la organización.

Como cierre al tema de los puntos de vista de la ingeniería, indicar que el uso de estos es de gran importancia por dos motivos:

1. Se puede dar el caso de que los ingenieros tengan experiencia en sistemas semejantes, por lo que pueden aportar requisitos en base a dicha experiencia.
2. Los técnicos que mantienen y administran el sistema, pueden sugerir requisitos que reduzcan el soporte de este. Indicar, que cada vez más los requisitos de administración son más importantes, ya que los costes de administración suponen un porcentaje elevado del coste total del sistema.

Entrevistas

En el proceso de obtención de requisitos, las entrevistas tienen un papel importante. Las personas encargadas de la obtención de estos, en este caso el equipo de la ingeniería de requisitos, mediante un conjunto de preguntas a los diferentes stakeholders sobre el sistema en uso y el sistema a desarrollar, obtendrán parte de los requisitos.

Existen dos tipos de entrevistas, abiertas y cerradas, donde la principal diferencia entre ambas radica en la definición o no definición previa de las preguntas a realizar. Es decir, en una entrevista cerrada el conjunto de stakeholders responde a una batería de preguntas predefinidas con anterioridad, mientras que en una entrevista abierta, no existe predefinición de preguntas, sino que el equipo de ingeniería de requisitos junto con los stakeholders analiza una serie de temas de donde se obtiene una comprensión de las necesidades actuales. Habitualmente, en la realidad, las entrevistas suelen ser una combinación de ambas, donde ciertas preguntas predefinidas, dan lugar a otro grupo de preguntas no predefinidas en un principio.

La información obtenida en las entrevistas, añade información extra a la obtenida por otros medios como por la observación a usuarios, estudios de documentación, etc., ya que siempre en una entrevista hay información esencial que se ignora.

Para ser un buen entrevistador, efectivo, se han de tener dos peculiaridades, las cuales se mencionan a continuación:

- ❖ Incitan al entrevistado a comenzar el diálogo bien con una sugerencia de requisito, o bien con una pregunta. Se considera una forma más efectiva de obtener información útil de esta manera que preguntando directamente lo que se quiere.
- ❖ Están preparados para escuchar a los stakeholders, y en caso de que estos indiquen requisitos asombrosos, no tienen inconveniente en cambiar su parecer acerca del sistema. Evitan ideas meditadas y no tienen prejuicios.

Escenarios

Los escenarios son utilizados para proporcionar información adicional a la descripción de los requisitos. En la mayoría de los casos, a las personas les es más sencillo aportar ejemplos de la vida real, en lugar de ejemplos teóricos. Así como también les resulta más sencillo opinar y entender un escenario debido a que en estos se observa cómo se interactúa con el sistema. Cada escenario suministra distinto tipo de información debido a que cada uno de ellos engloba una o varias interacciones factibles. A continuación se detallan algunos aspectos que puede contener un escenario:

1. Explicación del flujo normal de eventos en el escenario.
2. Explicación detallada de lo que esperan el sistema y los usuarios cuando el escenario empieza.
3. Información adicional acerca de otras actividades que se podrían llevar a cabo.
4. Información acerca de la situación del sistema cuando el escenario finaliza.
5. Explicación de lo que puede no ir correctamente y como dirigirlo en ese supuesto caso.

Prototipo

A menudo, los clientes del nuevo sistema informan acerca de las características u objetivos que este debe de poseer, pero no indican de forma minuciosa los requisitos, es por ello, que en ciertas ocasiones los desarrolladores tienen dudas acerca de ciertos aspectos como por ejemplo la forma de interactuar entre la máquina y el ser humano o la eficiencia de algoritmo entre otras.

Por estas y otras razones probablemente el paradigma de hacer prototipos ofrezca un mejor enfoque. Es una brillante idea el uso de estos cuando los requisitos aún no están claros, ya que ayuda a aumentar la asimilación de lo que se va a producir.

El paradigma de hacer prototipos, como se puede observar en la ilustración 17 se compone de varias fases como son comunicación, plan rápido, modelado del diseño rápido, construcción del prototipo y despliegue, entrega y retroalimentación. A continuación se explican las diferentes fases, para tener una idea más detallada de las acciones que se realizan en cada una de ellas.

- *Comunicación:* es la primera fase del paradigma de hacer prototipos, donde se reúnen el conjunto de participantes, detallan los objetivos globales, reconocen los requisitos conocidos y localizan las áreas donde es indispensable una mayor definición.
- *Plan rápido:* en esta fase se proyecta velozmente una iteración para realizar el prototipo.
- *Modelado del diseño rápido:* llegados a este punto se lleva a cabo el modelado, cuya prioridad es representar los diferentes factores del software que serán perceptibles para los usuarios finales.
- *Construcción del prototipo:* como su nombre indica se construye el prototipo en base a los acuerdos alcanzados en las fases anteriores.
- *Despliegue, entrega y retroalimentación:* el prototipo es transmitido y valorado por los diferentes participantes, los cuales aportan un feedback con el fin de mejorar o corregir los requisitos.

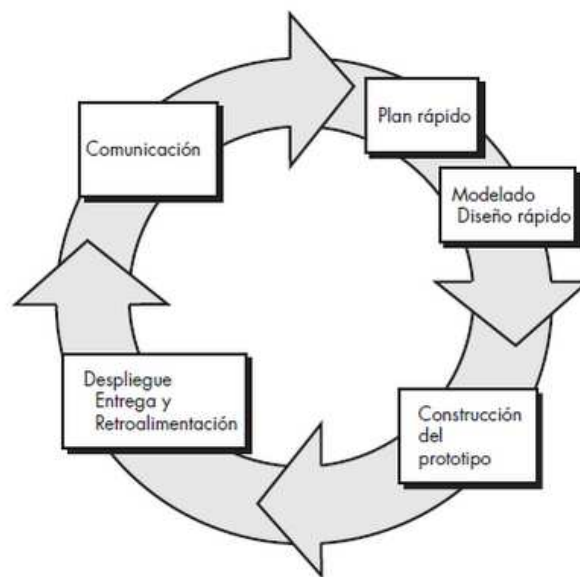


Ilustración 17: Proceso de realización de prototipos tomado de [Pre10]

Tanto a ingenieros como a usuarios les agrada la idea del paradigma de hacer prototipos, ya que por una parte los usuarios experimentan la emoción del sistema real y los ingenieros consiguen construir algo pronto, aunque en la mayoría de las ocasiones esto conlleva algunos problemas, pero no obstante la utilización de prototipos es una manera eficaz de obtener requisitos, y para que todo resulte más sencillo, basta con fijar las reglas del juego desde el inicio.

Lo perfecto sería que el prototipo fuera utilizado para extraer los requisitos software. Hay que tener en cuenta que si se va a construir un prototipo, para ello pueden ser reutilizados fragmentos de código ya existente o utilizar herramientas disponibles en el mercado que producen programas que funcionan. Además, en la mayoría de los casos nunca el primer prototipo creado es de gran utilidad, luego lo que se debe de hacer es crear posteriores versiones con mejoras introducidas que resuelvan los problemas iniciales.

Casos de uso

Un caso de uso detalla la forma en la que una persona o actor interactúa con el sistema mediante usos de escenarios y con el fin de obtener requisitos.

A continuación se muestra un ejemplo de caso de uso, con el fin de visualizar el conjunto de información que engloban estos, así como también la forma en la que está distribuida esta. Cabe mencionar, que existen diferentes formas de definir un caso de uso, algunas de ellas sencillas, en las que únicamente aparece información acerca del nombre, actores, objetivo, precondiciones y poscondiciones, pero sin embargo existen otros modelos muchos más completos, como el que se muestra a continuación:

Nombre:	(Nombre del caso de uso)
Autor:	(Nombre del autor o autores del caso de uso)
Fecha:	(Fecha creación del caso de uso)
Descripción:	(Corta descripción del caso de uso)
Actores:	(Actores/personas que participan en el caso de uso)
Precondiciones:	(Condiciones que deben cumplirse para poder ejecutar el caso de uso)
Flujo Normal:	(Situación normal de ejecución del caso de uso) (Happy day)
Flujo Alternativo:	(Flujos alternativos de ejecución del caso de uso)
Poscondiciones:	(Condiciones que deben cumplirse una vez que ha finalizado la ejecución del caso de uso)

Tabla: Ejemplo de caso de uso

Etnografía

La etnografía se basa en una técnica de observación, donde un analista se introduce en el ambiente laboral, con el fin de observar el trabajo diario y anotar las tareas reales de cada uno de los usuarios. El principal objetivo de la etnografía consiste en comprender los requisitos sociales y organizacionales. A la mayoría de la gente, les supone gran dificultad expresar los detalles del trabajo que realizan, ya que ellos lo ven como cosas normales y naturales, ellos entienden su propio trabajo pero no la relación con el resto de funciones de la entidad.

La etnografía es muy eficaz para adivinar dos tipos de requisitos específicos: *los requisitos que se derivan de la forma en la que la gente trabaja realmente (en lugar de la forma en la que se debería trabajar según procedimientos o según la definición de los procesos)* y *los requisitos que se derivan de la cooperación y conocimiento de las actividades de los usuarios.*

Puesto que se ha hablado anteriormente de prototipos, cabe señalar que a menudo estos se pueden combinar con la etnografía. Por una parte la etnografía proporciona información para la elaboración del prototipo, y por otra la elaboración de prototipos se basa en la etnografía con el fin de reconocer cuestiones y preguntas que puedan ser tratadas con el etnógrafo más tarde.

Introspección

La introspección es el método más claro que puede ser utilizado cuando se trata de comprender cuales son las características que debe poseer un sistema para que sea exitoso. Este

método tiene la ventaja de ser muy útil, pero está la desventaja de que un experto en un campo diferente, como la ingeniería de requisitos, es poco factible que manifieste la experiencia de los usuarios reales.

7.4 Descripción de requisitos

Según lo mencionado en apartados anteriores, uno de mayor peso del documento de especificación de requisitos consiste en la descripción y clasificación de estos. Como se sabe, dichos requisitos son utilizados por un amplio grupo de personas, los cuales utilizarán estos entre otras cosas para el diseño del sistema, es por ello, que la descripción juega un papel muy importante.

Cada requisito ha de ser definido individualmente y es aconsejable y deseable que tenga un conjunto de características, esto es, que sea *claro*, es decir que tenga un alto grado de comprensibilidad, *breve* y *no ambiguo*.

Como estos tienen que ser leídos por un gran número de personas, es recomendable utilizar conjuntos de palabras en lenguaje natural así como también diagramas. Como se indicó en el apartado “tipos de requisitos” ha de utilizarse este tipo de lenguaje, ya que aun no teniendo características deseables, como que es un lenguaje ambiguo, opaco y en ocasiones da lugar a malos entendidos, en general todo el mundo es capaz de leerlo, por lo que será el utilizado para la descripción de requisitos.

Probablemente se pueda pensar, que utilizando otro tipo de lenguaje, los problemas ocasionados por la utilización del lenguaje natural a la hora de realizar la descripción de los requisitos se podrían evitar, pero esto no es cierto, por lo que siempre se intentará realizar una buena escritura de estos usando un lenguaje lo más simple posible, evitar frases complejas y párrafos largos y utilizando términos técnicos solo cuando no se tenga otra opción.

Aunque no es muy apreciable, pero el hecho de que los requisitos estén escritos en lenguaje natural y estos sean entendidos por las partes interesadas sin dificultad, implica un ahorro en costes y tiempo, ya que si estos son fáciles de entender, el tiempo empleado en explicárselos a las partes involucradas es menor luego hay ahorro de tiempo, así como también las personas a las que se les paga para que lean los requisitos, reducen el tiempo en la lectura, luego también hay un ahorro de costes.

El escribir claro y breve, que a simple vista parece muy sencillo, en ocasiones no lo es, ya que para mucha personas eso es una tarea de gran dificultad, ya que estos tienden a escribir frases muy largas y con verbos no apropiados, lo que conlleva a expresar los requisitos de una manera no correcta. Como solución a esto, a menudo se propone crear una mini-guía (con una media de 4 o 5 páginas) que indique de forma resumida la forma de utilizar el lenguaje para la descripción de los requisitos en la organización.

Algunos ejemplos de pautas que podrían ser añadidas en la guía pueden ser los siguientes.

- Párrafos cortos, con una media de siete líneas aproximadamente.
- Poner especial atención a la ortografía y gramática, ya que una ortografía y gramática errónea puede conllevar a tergiversar el significado de un requisito.

- Frases cortas. Ya que si analizamos nuestra propia experiencia se puede observar que cuando se trata de frases muy largas, nos vemos en la necesidad de tener que leerlas varias veces para poderlas entender.
- Evitar el uso de abreviaturas y siglas, al menos que se esté al 100% seguro de que todos los lectores lo entenderán. Por ejemplo en este proyecto son innumerables las veces que se menciona el conjunto de palabras “documento de especificación de requisitos software”, cuando se podría poner de forma abreviada ERS, pero debido a que es un proyecto largo y se puede dar el caso de que todos los lectores no se percaten a lo que me puedo referir escribiendo ERS, he optado por poner toda la traducción.
- En el supuesto caso de tener que describir una relación compleja, siempre nos debemos de inclinar a la utilización de diagramas en lugar de intentar transmitir la idea en lenguaje natural, ya que el uso de estos es más eficaz y sencillo para transmitir este tipo de ideas.
- En caso de tener que utilizar referencias, mencionar aparte del número de referencia también aportar una breve descripción de a lo que se refiere, ya que, esto aporta al lector un refresco de memoria, pudiéndose dar el caso de recordar a lo que se refiere la mención y no verse obligado en todos los casos a tener que buscar dicha referencia con el fin de recordar a que hacía referencia dicho número.
- A menudo, en la descripción de los requisitos se suelen utilizar palabras/verbos como “imprescindible”, “deberá”, “debería” entre otros. Tener especial cuidado en el uso de estos y utilizarlos de una manera correcta. A continuación se indica el significado que dichas palabras deben tener:
 - Deberá: indica que el requisito es obligatorio.
 - Imprescindible: es mejor evitar el uso de esta palabra, pero en el caso de ser usada debe de tener el mismo significado que el verbo “deberá”.
 - Debería: indica que el requisito es esperado, pero que no es obligatorio.
- Utilizar la terminología correctamente, es decir, que el significado de una palabra sea el mismo en todas las partes del documento donde es utilizado, es decir, que no en un primer momento tenga un significado y las sucesivas veces tenga otro, siempre tiene que ser consistente.

Esta mini-guía que se comenta que es necesario crear con el fin de que los requisitos estén correctamente descritos, implica inicialmente un gasto de tiempo, ya que por una parte se necesita redactar, y por otra se necesita dar una pequeña formación a los implicados con el fin de que la conozcan, y también implica un aumento de costos, ya que esta necesita ser revisada. En mi opinión dicha guía a la larga es beneficiosa para la compañía, ya que todo el tiempo/dinero que en un principio se gasta, a la larga se recupera, ya que el tiempo empleado durante la descripción del conjunto de requisitos es menor.

La escritura de los requisitos en mi opinión tiene un grado de dificultad alto y es muy importante, ya que en mi caso en muchas ocasiones tengo que realizar pruebas de un determinado sistema en función de unos requisitos previos definidos para verificar que estos se cumplen y validar si la funcionalidad de dicho sistema es la esperada. Pues bien, la tarea más complicada de realizar en la mayoría de las ocasiones, consiste en comprender los requisitos dados, es decir, entender que es lo que expresa cada uno de los requisitos para saber cuál es la funcionalidad que hay que probar en el sistema. Esto es debido a que la persona encargada de la definición de estos, da por hecho que las

personas encargadas de realizar las pruebas parten del mismo conocimiento que él, cosa que es incierta.

Es por esto, que a continuación se mencionan algunos aspectos que hay que tener en cuenta a la hora de realizar la redacción de los requisitos:

- ❖ Siempre el número de veces que es leído el conjunto de requisitos es mucho mayor que el número de veces que son escritos estos, luego, es de gran importancia emplear el tiempo necesario con el fin de obtener un conjunto de requisitos que sean fáciles de leer y de entender. Ya que escribir estos de forma clara y precisa es complejo, luego ha de tomarse suficiente tiempo para redactar-revisar-mejorar tantas veces como sea necesario con el fin de que la especificación sea correcta.
- ❖ El escritor siempre tiene que suponer que el conjunto de personas que leerán los requisitos no tienen ningún conocimiento previo en el tema. Luego los tiene que redactar de tal manera que estos sean comprendidos por cualquier persona que los lea.

Por otra parte, es cierto que depende del tamaño del sistema, pero normalmente el número de requisitos que han de ser escritos es elevado, es por ello, que una de las cosas que se recomienda es crear una plantilla estándar (en el caso de que la organización no disponga de ella) para el conjunto de estos, la cual englobará un conjunto de campos con el fin de que en ellos esté contemplada toda la información necesaria para la especificación de cada uno de los requisitos.

La utilización de esto implica un conjunto de ventajas como por ejemplo *facilidad de lectura*, ya que, una vez que los lectores han comprendido y entendido la plantilla, el entender el resto de requisitos es mucho más sencillo, y el tiempo empleado es mucho menor que si no se utilizan. Por otra parte *facilidad en la escritura*, ya que una vez conocida la información que recoge dicha plantilla, la escritura es mecánica, además la probabilidad de olvido de información es menor y la revisión de los requisitos (y si figura otras veces) más sencilla. Y por último *requisitos más fáciles de recoger*.

Como se ha indicado en párrafos anteriores, para la descripción de requisitos aparte del lenguaje natural, también es recomendable el uso de diagramas. Los diagramas principalmente serán utilizados siempre y cuando se necesite transmitir información estructurada, relaciones complejas o la relación existente entre información en la descripción de requisitos.

El uso de diagramas proporciona una serie de beneficios a los lectores cuando estos son utilizados. Algunos de estos beneficios se mencionan a continuación:

- ✚ Una vez que algunos diagramas han sido utilizados en el documento de especificación de requisitos, estos a su vez pueden reutilizarse cuando se necesita hacer presentaciones para los clientes acerca de requisitos.
- ✚ Son más sencillos de entender que las descripciones de texto. Si nos paramos a analizar nuestro comportamiento, observaremos que siempre que queremos explicar algo, se tiende a utilizar diagramas para ello, en lugar de escribir una parrafada, ya que el mostrarlo mediante un “dibujo” favorece la comprensión.
- ✚ La introducción de un diagrama proporciona dos ventajas. La primera ventaja consiste en que mediante el uso del diagrama la información es más clara y por otra parte, otra de las

ventajas es que el texto de alrededor es más fácil de leer ya que se presenta en fragmentos más pequeños y legibles.

Llegados a este punto, se sabe que para la descripción de requisitos se debe utilizar texto y también diagramas, pero probablemente nos preguntemos ¿Cuándo se deben utilizar los diagramas? Bien, pues a continuación se indican los casos en los que es más aconsejable utilizar diagramas:

- Cuando el sistema que queremos representar está compuesto por un conjunto de módulos y el objetivo es mostrar la relación existente entre cada uno de ellos.
- Cuando se necesita mostrar la descomposición de un módulo.
- Cuando se necesita mostrar un conjunto de actividades, donde cada una de ellas está compuesta por un grupo de tareas, y donde cada tarea tiene sus correspondientes entradas y salidas.

Capítulo 8: Tipos de requisitos

Normalmente el conjunto de requisitos software se clasifican en requisitos funcionales, requisitos no funcionales y requisitos del dominio. Todos ellos son requisitos del sistema, pero ¿Cuál es la diferencia entre ellos? Pues bien, antes de estudiar a fondo cada uno de ellos, se dará una pequeña descripción para tener una idea general.

- Requisitos funcionales: contienen información relativa a los servicios que el sistema debe prestar. Además en muchas ocasiones también engloban aspectos relativos a lo que el sistema no debe hacer.
- Requisitos no funcionales: Se aplican al sistema en su totalidad, y estos reflejan las limitaciones de las tareas ofrecidas por el sistema.
- Requisitos del dominio: pueden ser funcionales o no funcionales y manifiestan propiedades y limitaciones del dominio del sistema.

Así, a simple vista, una vez que han sido leídas las diferentes definiciones de los distintos tipos de requisitos, parece muy obvia la clasificación, pero en la realidad no es así, ya que en repetidas ocasiones dan lugar a confusión.

8.1 Requisitos funcionales

Bien, como se ha mencionado anteriormente, los requisitos funcionales detallan como el sistema debe actuar, es decir, que es lo que este debe de hacer. Es por ello que estos detallan entradas, salidas y particularidades entre otros.

Por otra parte la especificación de requisitos funcionales debe de poseer ciertas características, debe ser completa y consistente. Para que sea completa y consistente debe de cumplir la especificación de requisitos las siguientes características:

- Completa: todos los posibles servicios demandados por el usuario están definidos.
- Consistente: Cuando no existen contradicciones entre el conjunto de requisitos, es decir, las definiciones del conjunto de requisitos son coherentes.

Señalar, que cuando nos encontramos con sistemas de gran tamaño y gran complejidad, el llevar a cabo que la definición de requisitos sea completa y consistente, es casi imposible por las siguientes razones:

- ✚ Cada uno de los stakeholders tiene distintos tipos de necesidades, y en ocasiones opuestas.
- ✚ La especificación de requisitos para sistemas complejos y grandes, la mayoría de las veces implica con gran facilidad cometer fallos y olvidos.

8.2 Requisitos no funcionales

Cuando se habla de requisitos no funcionales, se refiere al conjunto de estos que especifican las propiedades emergentes del sistema como disponibilidad, tiempo de respuesta, rendimiento del sistema y capacidad de almacenamiento entre otras y aparecen por diferentes motivos, entre ellos políticas de la organización, necesidades de usuario y restricciones de usuario entre otros.

Cabe señalar, que dicho tipo de requisitos son de gran importancia, ya que el hecho de que un requisito funcional no se cumpla, es un problema, pero en menor medida que si se da el caso de que el requisito que no se cumple es un requisito no funcional. Esto es debido a que cuando un requisito funcional no se cumple, el usuario del sistema puede encontrar formas alternativas de trabajar alrededor de la función del sistema que no cumple la necesidad deseada, pero cuando se trata de un requisito no funcional, puede llegar a darse el caso extremo de que el sistema no sea útil.

Debido a la gran cantidad de tipos de requisitos no funcionales existentes, seguidamente se muestra una clasificación de ellos:

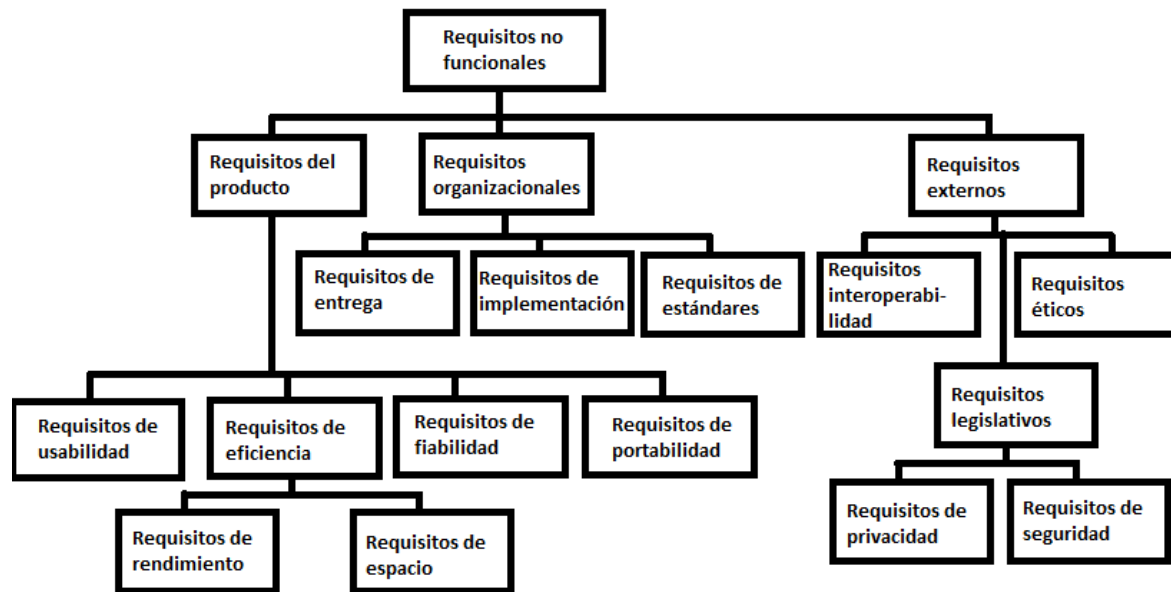


Ilustración 18: Clasificación requisitos no funcionales tomados de [Som05]

Aunque el nombre de cada grupo de requisitos es muy intuitivo, no obstante, a continuación se muestra una breve descripción de los grupos raíz:

- ❖ Requisitos del producto: este tipo de requisitos detallan el modo de actuación del producto. A su vez estos engloban requisitos de usabilidad, requisitos de eficiencia, requisitos de fiabilidad y requisitos de portabilidad.

Para clarificar mas más este tipo de requisitos, ponemos un ejemplo de requisito del producto: "A la hora de realizar una consulta en el sistema, este debe mostrar el resultado de la ejecución en un tiempo inferior a dos minutos"

- ❖ Requisitos organizacionales: estos requisitos provienen de procedimientos y políticas presentes en la organización del desarrollador y cliente. A su vez, los requisitos organizacionales engloban requisitos de entrega, requisitos de implementación y requisitos de estándares.

Un ejemplo de este tipo de requisito sería "Cada uno de los entregables que se realicen a lo largo del desarrollo del sistema serán realizados siguiendo lo concretado en Métrica V3".

- ❖ Requisitos externos: este grupo de requisitos probablemente incluya al mayor número de estos debido a que engloba al conjunto de requisitos que proceden de los factores externos al sistema y de su proceso de desarrollo. A este grupo pertenecen los requisitos de interoperabilidad, requisitos éticos y por último los requisitos legislativos.

Como ejemplo de este tipo, un requisito externo sería: "El sistema debe registrar información del tipo hora y usuario introducido entre otros cuando se realiza un acceso erróneo o no válido al sistema".

Uno de los problemas conocidos respecto a los requisitos no funcionales consiste en la dificultad de probar o verificar estos, ya que a menudo, los usuarios del sistema a desarrollar, informan de requisitos no funcionales de manera ambigua como capacidad de recuperación de fallos, o facilidad de uso lo que origina disputas y polémicas una vez que el sistema ha sido entregado al cliente. Es por esto, que siempre que se pueda, deben de ser redactados estos de forma cuantitativa para que pueda ser probado de una forma objetiva, el inconveniente es que esto no es siempre fácil para los clientes de un sistema

Y para finalizar el apartado de requisitos no funcionales, informar que hay que prestar especial atención en ellos, ya que muy a menudo estos entran en conflicto con otros requisitos (funcionales o no funcionales).

8.3 Requisitos del usuario

La definición de los requisitos de usuario consiste en explicar requisitos funcionales y no funcionales para que sean comprendidos por los usuarios del sistema sin poseer estos un conocimiento técnico. Es por esto, que a la hora de redactar dichos requisitos, han de utilizarse tablas y diagramas sencillos así como también un lenguaje sencillo. Además hay que tener en cuenta que solamente se debe detallar la forma de actuar externa del sistema y no las características de diseño del problema siempre y cuando sea posible.

El objetivo de redactar los requisitos mediante lenguaje cotidiano, en ocasiones genera algunos problemas, los cuales son mencionados a continuación:

- Equivocación de requisitos: debido a que no se diferencian claramente la información existente para el diseño, los requisitos funcionales y no funcionales y los objetivos del sistema.
- Falta de claridad: en ocasiones es difícil usar el lenguaje de forma rigurosa y no ambigua sin realizar el documento poco breve y complicado de leer.
- Unión de requisitos: varios requisitos distintos pueden ser declarados como un solo requisito.

Los requisitos del usuario que engloban demasiada información, por una parte son complicados de entender y por otra limitan a los desarrolladores del sistema en cuanto a aportar posibles soluciones a los problemas de usuario, por ello dichos requisitos deben centrarse en los recursos principales que deben ser proporcionados. No obstante, es una buena práctica el añadir junto a cada requisito un razonamiento o motivo que informe el porqué de la inclusión de dicho requisito a la lista de requisitos de usuario.

Existen un conjunto de medidas conocidas para reducir los posibles malos entendidos a la hora de escribir los requisitos de usuario, a continuación se encuentran algunos de ellos:

- ✓ Emplear el lenguaje de forma consistente. Esto es, en todo momento hay que diferenciar entre requisitos obligatorios y requisitos deseables. Los requisitos deseables son aquellos que no son imprescindibles y por ello han de ser redactados en el tiempo futuro condicional y los requisitos obligatorios son los requisitos fundamentales y a los que el sistema debe dar soporte y por ello han de ser redactados en futuro simple.
- ✓ Prescindir siempre que se pueda del uso de vocablos técnicos informáticos en los requisitos de usuario.
- ✓ En el caso de existir palabras claves referentes al requisito, destacar estas de alguna manera, ya sea con cursiva, negrita o color.
- ✓ Definir un formato estándar a utilizar para todos los requisitos, y verificar que todos ellos están definidos según este. Esto implica que los olvidos sean menores así como también facilidad a la hora de la verificación de los requisitos. El formato utilizado presenta el requisito en negrita, incluyendo un detalle del fundamento con cada requisito de usuario y una referencia al detalle de los requisitos del sistema. Además también se puede incluir información adicional como nombre de la persona que realizó la propuesta, para que en el caso de posibilidad de cambio se conozca a la persona a la que acudir.

8.4 Requisitos del sistema

Existe una versión ampliada y detallada de los requisitos de usuario, que se corresponden con los llamados requisitos del sistema y son usados por los ingenieros de software como inicio para el diseño del sistema. En estos básicamente se especifica la manera en que el sistema debe proveer los requisitos del usuario.

En ningún momento los requisitos del sistema detallan como debe ser el diseño ni la implementación del sistema, sino que informan acerca del comportamiento y restricciones operativas de este. En cuanto a la forma de redactar dichos requisitos debe utilizarse un lenguaje natural igual que ocurre con los requisitos de usuario, pero aun siendo este el objetivo no siempre es sencillo y puede provocar confusión y dificultad de comprensión por diversos motivos que son mencionados a continuación:

- Cuando se utiliza un lenguaje natural a la hora de realizar la especificación de requisitos, conlleva que esta sea bastante flexible, lo que quiere decir que se pueda decir lo mismo de diferentes formas, es por esto que se deja camino libre al lector para interpretar cuando los requisitos son los mismos y cuando son diferentes.
- Lo ideal sería tener modularizados el conjunto de requisitos, de forma que a la hora de realizar un cambio, no se tuviera que revisar cada uno de ellos, sino ir directamente al grupo afectado, pero lo que sucede es que esto no es fácil de realizar.

- En lenguaje natural y para la especificación de requisitos, para que la comprensión sea la correcta, redactores y lectores deben de usar las mismas palabras para el mismo concepto, ya que de no ser así puede llevar a confusión y a erróneas interpretaciones debido a la ambigüedad existente.

Como consecuencia de los problemas anteriores se tienen malas interpretaciones de la especificación, que normalmente son reveladas en fases posteriores del proceso de software y cuya solución puede llegar a ser muy costosa.

Como se mencionó previamente, es primordial escribir los requisitos de usuario en lenguaje natural de manera que las personas sin conocimiento alguno pueden comprender dicha especificación. Existe un conjunto de notaciones para la especificación de requisitos que se resume en el cuadro siguiente:

NOTACION	DESCRIPCIÓN
Lenguaje natural estructurado	Este enfoque depende de la definición de formularios o plantillas estándares para expresar la especificación de requisitos.
Lenguajes de descripción de diseño	Este enfoque utiliza un lenguaje similar a uno de programación, pero con características más abstractas, para especificar los requisitos por medio de la definición de un modelo operativo del sistema. Este enfoque no se utiliza ampliamente en la actualidad, aunque puede ser útil para especificaciones de interfaces.
Especificaciones matemáticas	Para definir los requisitos funcionales del sistema, se utiliza un lenguaje gráfico, complementado con anotaciones de texto. Uno de los primeros lenguajes gráficos fue SADT (Ross, 1977)(Schoman y Ros, 1977). Actualmente, se suelen utilizar las descripciones de caso de uso y los diagramas de secuencias.
Notaciones gráficas	Son notaciones que se basan en conceptos matemáticos como el de las máquinas de estado infinito o los conjuntos. Estas especificaciones no ambiguas reducen los argumentos sobre la funcionalidad del sistema entre el cliente y el contratista. Sin embargo, la mayoría de los clientes no comprenden las especificaciones formales y son reacios a aceptarlas como un contrato del sistema

Tabla 12: Notaciones a poder utilizar en la ERStomado de [Som05]

8.5 Requisitos del dominio

Los requisitos del dominio proceden del dominio de la aplicación y no de las necesidades concretas de los usuarios y estos son de gran importancia ya que con frecuencia manifiestan los fundamentos del dominio de aplicación.

Capítulo 9: Modelo CMMI

9.1 Introducción al modelo CMMI

El principal objetivo cuando se está realizando el desarrollo de requisitos consiste en extraer y analizar información con el fin de constituir los diferentes tipos de requisitos: requisitos del producto, requisitos del cliente y requisitos de componente del producto.

Según lo indicado en el párrafo anterior, se deduce la existencia de tres tipos de requisitos, los del cliente, los del producto y los del componente del producto, que todos ellos unidos estudian y analizan las propiedades del producto así como también las exigencias de las principales partes involucradas, englobando las necesidades referentes a las distintas etapas del ciclo de vida.

Se parte de la base, de que para todo proyecto se ha de desarrollar el conjunto de requisitos pertinentes, ya que estos, serán el pilar principal para la realización del diseño. La fase de desarrollo de requisitos engloba las siguientes actividades:

- Con respecto a las partes interesadas, análisis de las diferentes necesidades de estas.
- Con respecto al producto.
- Extracción, estudio.

Debido a que el cliente puede aportar requisitos con particularidades interesantes acerca del diseño, se obtendrán requisitos de producto, así como también requisitos de cliente. En etapas sucesivas, el conjunto de los requisitos de cliente, serán depurados, pasando a formar parte de los requisitos del producto y requisitos de componentes de producto.

A lo largo del ciclo de vida del producto, serán reconocidos nuevos requisitos así como también los ya existentes serán depurados, con el fin de tomar las medidas correctivas necesarias. El conjunto de decisiones tomadas serán estudiadas para conocer cuál será la repercusión sobre los requisitos.

La etapa de desarrollo de requisitos engloba tres objetivos concretos:

1. *Desarrollar requisitos del cliente*: especificación de los requisitos del cliente, los cuales serán usados para la especificación de los requisitos del producto.
2. *Desarrollar requisitos del producto*: especificación de los requisitos del producto o componentes del producto, con el fin de ser usados para el diseño del producto o de componentes del producto.
3. *Estudio y validación de requisitos*: estudia los tres grupos de requisitos definidos anteriormente con el objetivo de especificar, deducir y entender el conjunto de requisitos. La consecución de esta meta, ayudará a la consecución de las dos metas anteriores.

Como se puede estar observando, se usa mucho el análisis, pero ¿Para qué es utilizada dicha técnica? Bien, pues principalmente los análisis son usados para entender e interpretar, explicar y elegir los requisitos a partir de competitivas opciones. En general, un análisis engloba lo siguiente:

- ✓ Explicación de cuál es la función del producto en términos de limitaciones de diseño, accesibilidad, grado de adaptación y capacidad de respuesta entre otros.
- ✓ Estudio con el fin de conocer los requisitos y saber que se necesita para cada una de las etapas del ciclo de vida englobando a las principales partes afectadas, factores como seguridad y factibilidad entre otros y el entorno de operación.
- ✓ Desarrollo de una noción operacional (entre otros conservación, retiro y soporte).

Como se está hablando a lo largo de esta sección del desarrollo de diferentes tipos de requisitos así como también del análisis y validación de estos, a continuación se tratarán con el fin de conocerlos en profundidad.

9.2 Desarrollo de requisitos del producto y requisitos de componente de producto

A continuación se formulan un conjunto de preguntas con sus correspondientes respuestas, todo ello con el fin de que sea más estructurado y más fácil de comprender en que se basa el desarrollo de requisitos del cliente.

¿A quién se refiere cuando se habla de las partes interesadas?

Bien, pues para recordar, las partes interesadas en un proyecto serán las personas o conjunto de personas que de alguna manera estarán involucrados en el mismo o tendrán algún tipo de repercusión por parte de este, ya sea por disfrutar de algún beneficio proveniente de dicho proyecto, o por el contrario obtener algún perjuicio. Cada organismo tiene un conjunto de partes interesadas predefinidas, las cuales intervendrán en el proyecto directa o indirectamente. Este conjunto de personas son las que se conocen como “stakeholders”, ya estudiados en secciones anteriores.

Existen dos tipos de partes interesadas, las partes interesadas activas y las pasivas. La diferencia entre ambas radica en que las partes interesadas pasivas, se ven perjudicadas por el proyecto pero indirectamente, es el caso por ejemplo de los familiares, mientras que las partes interesadas activas colaboran de forma directa en el proyecto, como es el caso del director y jefes de proyecto entre otros.

¿Cuál es la base principal para establecer los requisitos del cliente? ¿Que se estudia para la obtención de estos?

El principal aspecto a estudiar con el fin de obtener los requisitos del cliente son las necesidades de los diferentes stakeholders.

Dentro de toda la información analizada incluyendo como se ha indicado las necesidades, además cabe destacar los intereses, pensamientos acerca del producto, limitaciones e interfaces de los stakeholders. Todo este conjunto de información es estudiado, arreglado, clarificado y preparado para transformarlo en requisitos del cliente. Uno de los objetivos a cumplir a lo largo de toda la vida del proyecto es que el conjunto de información anterior, es decir limitaciones, interfaces, intereses e ideas acerca del producto, debe entenderse y reconocerse claramente. Para que la interacción necesaria sea más fácil, a menudo se implica a un suplente del usuario final con el fin de que este solucione posibles problemas así como también reproduzca las necesidades de dichos usuarios.

¿En qué consiste la deducción de las necesidades?

El verbo “educir” no solo engloba la obtención de requisitos a través de la identificación de requisitos agregados y no aportados por los clientes. Este conjunto de requisitos adicionales a los que nos referimos deben tratar por una parte los impactos en el producto y por otra las tareas del ciclo de vida del producto.

¿Qué técnicas existen para la educación de las necesidades?

Existen numerosas técnicas con el fin de deducir el conjunto de necesidades de las partes interesadas. A continuación se muestra un listado, de las cuales sobre algunas de ellas ya se ha hablado:

- Demostraciones de tecnología
- Grupos de trabajo de control de la interfaz
- Grupos de trabajo de control técnico
- Revisiones intermedias del proyecto
- Cuestionarios, entrevistas y escenarios obtenidos de los usuarios finales
- Walthroughs de soporte, desarrollo y de operación, y análisis de tareas de usuarios finales
- Workshops con las partes interesadas para la educación de los atributos de calidad.
- Prototipos y modelos
- Tormenta de ideas
- Casos de uso
- Historias de usuario
- Pequeñas entregas incrementales “por rodajas” de la funcionalidad del producto
- Análisis de casos de negocio
- Estudios de mercado

¿Cuáles son las técnicas que engloba el desarrollo de los requisitos de cliente?

Las actividades que engloba son las siguientes:

- Se deben afianzar las diferentes entradas procedentes de las principales partes interesadas.
- Con el fin de disponer de toda la información, esta ha de ser recuperada.
- En caso de existir algún tipo de inconveniente/conflicto, este debe ser solucionado.

¿De dónde provienen el conjunto de requisitos?

Los requisitos de cliente pueden provenir de dos fuentes claramente diferenciadas:

- *Opción 1:* de parte del cliente, el cual aporta los requisitos que considera oportuno.
- *Opción 2:* Obtención de requisitos de un proyecto ya existente.

En muchas ocasiones, los requisitos de cliente pueden pasar a estar en conflicto con las necesidades, intereses, limitaciones e interfaces. En el supuesto caso de darse esto, el objetivo es llegar a obtener un conjunto de requisitos de cliente una vez solucionados todos los conflictos existentes.

¿Qué tipo de información puede ser incluida en los requisitos de cliente?

Los requisitos de cliente pueden comprender necesidades, intereses, limitaciones e interfaces con respecto a la validación y verificación.

9.2.1 Desarrollo de requisitos del producto y requisitos de componente de producto

¿De dónde son obtenidos los requisitos de producto y requisitos de componentes de producto?

Como una primera idea, partiendo de los requisitos de cliente del apartado anterior, se obtendrán los requisitos de producto y requisitos de componente de producto.

Por lo tanto, los requisitos de producto y requisitos de componentes de producto son más minuciosos e indispensables y serán deducidos a partir del análisis de los requisitos del cliente junto con el desarrollo del concepto de operación.

¿Cómo son establecidos los requisitos de producto y requisitos de componente de producto?

El conjunto de características del cliente referente a calidad, unido a los requisitos funcionales pueden manifestarse en términos de producto así como también pueden ser especificaciones no técnicas. Por lo tanto, los requisitos de producto son obtenidos de dichos requisitos anteriores mediante el uso de términos técnicos y estos podrán ser usados para tomar diferentes determinaciones acerca del diseño.

Diferentes propiedades del proyecto como efectividad y factibilidad, objetivos a conseguir en este, complacencia del cliente y el negocio, son temas que son estudiados y tratados por el conjunto de requisitos del producto y requisitos de componente del producto.

Otros tipos de necesidades referentes a diferentes etapas del ciclo de vida serán tratados en los requisitos derivados, dependiendo de la afinidad existente entre las necesidades y los objetivos del negocio.

9.3 Gestión de requisitos con CMMI

¿Cuál es la finalidad de la gestión de requisitos?

Como su nombre indica, el principal objetivo o finalidad de la gestión de requisitos es gestionar por una parte los requisitos de los productos y requisitos de componentes de producto y por otra garantizar la organización entre el conjunto de requisitos anteriores, planes y productos de trabajo de proyecto.

A continuación se mencionan algunas anotaciones con el fin de introducirnos un poco más en el tema de la gestión de requisitos:

Todo el conjunto de requisitos producidos o recibidos por el proyecto, incluidos tanto requisitos técnicos como requisitos no técnicos y requisitos impuestos por la organización al proyecto, son tramitados por el proceso de gestión de requisitos.

Si el área de proceso de desarrollo de requisitos es implementada, como consecuencia de esto se producirán requisitos de producto y requisitos de componentes de producto, los cuales igual que para los demás casos, serán tratados por los procesos de gestión de requisitos.

En la totalidad de las áreas, cuando es utilizada la expresión “componente de producto” o la palabra “producto”, en sus significados pronosticados se engloban los servicios, los sistemas de servicio y sus componentes.

Cuando existen áreas específicas implementadas, refiriéndose esto a la Gestión de requisitos, Desarrollo de requisitos y Solución técnica, los distintos procesos de cada una de estas, pueden ejecutarse de forma convergente debido a que sus procesos pueden estar relacionados.

El proyecto de forma sincronizada realiza el conjunto de pasos pertinentes con el fin de garantizar que el grupo de requisitos se gestiona para aportar apoyo al conjunto de necesidades de planificación y ejecución del proyecto.

En un proyecto, cuando se reciben un conjunto de requisitos provenientes de un proveedor de requisitos previamente aprobado, la dinámica seguida consiste en examinar estos junto con el proveedor con el principal propósito de solucionar las cuestiones y evitar malos entendidos antes de que dichos requisitos sean incorporados en los planes del proyecto. Cuando ambas partes, es decir, el proveedor y el destinatario de los requisitos llegan a un acuerdo, se alcanza un compromiso por parte de los participantes en el proyecto sobre los requisitos.

En la gestión de requisitos, una de las tareas que engloba dicha actividad consiste en documentar cada una de las modificaciones realizadas a los requisitos, así como también fundamentar el análisis razonado y mantener entre los requisitos fuente la trazabilidad bidireccional. Los cambios o modificaciones en los requisitos, en el caso de existir, pueden ser documentados de diferentes formas, es decir, como petición de cambio por parte del cliente o de usuarios finales o podrían tomarse como nuevos requisitos recibidos de la fase de desarrollo de requisitos.

GESTIONAR LOS REQUISITOS

Durante la vida de un proyecto, este mantiene un grupo aceptado y actual de requisitos, realizando las tareas que a continuación se mencionan:

- Tramitar las modificaciones realizadas a los requisitos.
- Sustentar las relaciones entre productos de trabajo, requisitos y planes de proyecto, así como también garantizar la alineación entre estos.
- Tomar decisiones acerca de acciones correctivas.

9.3.1 Interpretación de requisitos

Según se originan los requisitos así como también avanza y se desarrolla el proyecto, el conjunto de actividades recibirán requisitos. Se determinarán un conjunto de pautas con el fin de señalar los canales oficiales desde donde serán recibidos los requisitos, para evitar un chorro continuo de estos.

En el momento en el que se recibe un requisito, este es examinado con el proveedor con el fin de garantizar que se logra una comprensión concordante del significado de dicho requisito. La salida obtenida de dicho análisis, es un grupo admitido de requisitos.

9.3.2 Obtención del compromiso sobre los requisitos

Como su nombre indica, el principal objetivo de esta actividad consiste en alcanzar el compromiso de los colaboradores del proyecto sobre los requisitos. Es por esto que dicha actividad se encarga de realizar el conjunto de actividades esenciales con el fin de implementar los requisitos.

Según avanza el proyecto, los requisitos van evolucionando, y a medida que sucede esto, esta actividad garantiza que el conjunto de participantes o colaboradores del proyecto se responsabilizan con los requisitos (aceptados y actuales) y con las modificaciones provenientes de los planes, productos y actividades del proyecto.

9.3.3 Tramitar modificaciones a los requisitos

El conjunto de requisitos no son estáticos, sino que estos cambian por diversos motivos, esto es, según varían las necesidades y prospera el trabajo, la probabilidad de que se tengan que realizar modificaciones a los requisitos es alta. Es primordial tramitar este conjunto de cambios de una manera eficaz y hábil. Para estudiar con eficiencia el posible choque que pueden originar las modificaciones a realizar, es imprescindible saber acerca de la fuente de cada uno de los requisitos y también que esté documentado el razonamiento obtenido del análisis de cualquier modificación.

9.3.4 Sustentar la trazabilidad bidireccional de los requisitos

En una primera lectura, cuando leemos “trazabilidad direccional”, probablemente no se sepa a qué se refiere, es por ello que lo primero que se hará será definir dicho término:

Trazabilidad bidireccional: Una asociación entre dos o más entidades lógicas que es discernible en ambos sentidos (es decir, hacia y desde una entidad).

Cuando el conjunto de requisitos es tratado adecuadamente, puede determinarse la trazabilidad desde un requisito hasta sus requisitos de más bajo nivel y viceversa. Dicha trazabilidad en dos direcciones proporciona asistencia para establecer si la totalidad de los requisitos fuente han sido tratados completamente y si el conjunto de requisitos correspondientes a los niveles más bajos pueden ser proyectados hacia una fuente válida.

Hay que tener en cuenta que la trazabilidad es esencial para calcular el choque de las posibles modificaciones en los requisitos sobre los productos de trabajo y las diferentes actividades dentro del proyecto. Además, en ocasiones dicha práctica no está automatizada, es por ello que debe realizarse de forma manual mediante la utilización como por ejemplo de bases de datos o herramientas habituales existentes.

Entre muchos de los aspectos a estudiar en la trazabilidad, algunos de ellos son los siguientes:

- El alcance, es decir, analizar cuáles son los límites donde la trazabilidad es imprescindible.
- Debido a que existen diferentes tipos de trazabilidad, esto es trazabilidad horizontal y trazabilidad vertical, analizar cuando es esencial un tipo u otro.
- Descripción de trazabilidad: define el conjunto de elementos que precisan de relaciones lógicas.

9.3.5 Garantizar el alineamiento entre los requisitos del trabajo y el proyecto

Esta tarea concreta descubre las incoherencias entre productos de trabajo, planes y requisitos del proyecto, así como también emprende comportamientos correctivos con el fin de solucionarlos.

9.4 Análisis y validación de requisitos con CMMI

Como su nombre indica, en esta fase en la que nos encontramos, se analiza y valida el conjunto de requisitos obtenidos en las etapas anteriores (requisitos de cliente y requisitos de producto), con relación al ámbito que se prevé del usuario final y como consecuencia de esto, esta fase da apoyo por una parte al desarrollo de los requisitos del cliente y por otra al desarrollo de los requisitos del producto.

Los análisis son realizados con el fin de definir cuál será el impacto ocasionado del dominio de operación predicho sobre la habilidad para cumplir las necesidades, intereses, limitaciones y las interfaces del grupo de partes interesadas. Nunca deben olvidarse aspectos como limitaciones de coste, factibilidad, necesidad del cometido, volumen del posible mercado y la táctica de consecución. Además se determina una explicación acerca de la funcionalidad necesaria así como también de las propiedades de calidad.

Cuando un análisis es realizado, los principales propósitos que se persiguen son por una parte concretar el conjunto de requisitos postulantes para los conceptos del producto que cumplan las necesidades, intereses, y limitaciones de las partes interesadas y por otra parte interpretar estos conceptos en un conjunto de requisitos. Al mismo tiempo que se realizan estas actividades, el conjunto de valores considerados como indispensables y necesarios para valorar la eficiencia del producto serán decididos en función del concepto inicial del producto y las entradas del cliente.

Probablemente, nos podamos preguntar ¿Por qué se validan los requisitos cuando estos ya han sido estudiados minuciosamente cuando han sido creados? Pues bien, el conjunto de requisitos es verificado con el fin de aumentar la probabilidad de que el producto elaborado actúe según lo esperado en el dominio donde va a ser utilizado.

Debido a que la anterior explicación acerca del análisis y validación de requisitos ha sido muy escueta, en los apartados posteriores se entrará en detalle acerca de estas dos actividades.

9.4.1 Determinación de escenarios y conceptos de operación

Según se lee el título de esta sección, lo primero que nos podemos preguntar es ¿A qué se refiere cuando se habla de escenarios? Pues bien, un escenario es una sucesión de acontecimientos que pueden darse mientras el producto o proyecto es desarrollado, usado o mientras se da soporte a

este. Estos escenarios son usados para poner de manifiesto las propiedades de calidad o necesidades funcionales de las partes interesadas. Hay que tener en cuenta que los escenarios pueden realizarse con el fin de estudiar sucesiones de desarrollo y operación entre otros.

Por otra parte, ¿De qué depende un concepto de operación? Un concepto de operación para un determinado producto depende de dos factores, por un lado del escenario y por otro lado de la solución del diseño. Se ha de tener en cuenta que los conceptos de operación son perfeccionados según se van tomando decisiones acerca de las soluciones y se especifican los requisitos de niveles más bajos.

Los escenarios y conceptos de operación progresan con el fin de simplificar la elección de soluciones con respecto al componente del producto, que una vez implementado, simplifican el soporte y desarrollo del producto, así como también satisfarán el uso pronosticado de este.

Los escenarios y conceptos de operación contendrán información acerca de la interacción entre usuarios finales, entorno y componentes del producto con otros componentes del producto.

9.4.2 Determinación de la definición de funcionalidad y de las particularidades o atributos de calidad necesarios

Uno de los caminos a poder seguir con el fin de establecer la definición de funcionalidad y atributos de calidad necesarios, es estudiar escenarios mediante el uso de “análisis funcional”, con el fin de detallar que es lo que se requiere que el producto haga. Este proceso de detalle que se comenta puede abarcar desde entradas, sucesiones, salidas y comportamientos hasta cualquier otro tipo de información que notifique el modo en el que se usará el producto. La definición obtenida de las funciones, agrupaciones lógicas de funciones y la unión con el conjunto de requisitos, se designa como “arquitectura funcional”.

En cuanto a los atributos de calidad, muchos de los que aparecerán serán de gran valor para la arquitectura, y es por ello que encaminará el desarrollo de la arquitectura del producto.

9.4.3 Análisis de requisitos

Una vez definido el concepto y los escenarios de operación, teniendo en cuenta estos, el conjunto de requisitos correspondientes al nivel uno serán estudiados con el fin de averiguar si estos son esenciales y suficientes para satisfacer las metas de niveles superiores de la jerarquía del producto.

Los requisitos ya estudiados, subsiguientemente aportarán la base para la transformación a requisitos con mayor grado de detalle y exactitud en los niveles más inferiores de la jerarquía del producto.

Según se explican los requisitos, debe entenderse la correspondencia de estos con los requisitos de niveles posteriores así como también de las características de calidad necesarias. Además se identifican los requisitos, los cuales serán usados para hacer el seguimiento del progreso.

9.4.4 Análisis de requisitos con el fin de alcanzar un equilibrio

Se analizarán los requisitos para alcanzar un equilibrio entre las necesidades y las limitaciones de las partes interesadas.

Las necesidades y limitaciones de las partes interesadas referirán temas como costo, riesgo, calendario, productividad del proyecto, utilidad, preferencias, elementos reutilizables o facilidad de mantenimiento.

9.4.5 Validar los requisitos

De forma resumida, la actividad de validación de requisitos persigue verificar que el producto o proyecto satisface todas las necesidades, y como resultado este desempeña las actividades según lo esperado en el ámbito del usuario final.

La validación de los requisitos se lleva a cabo con los usuarios finales y en fases tempranas de desarrollo, con el fin de obtener confianza en que el conjunto de requisitos son adecuados para dirigir un desarrollo que proporcione como salida, una exitosa validación final.-

Por regla general, en determinados tipos de organizaciones, como por ejemplo cuando estas son maduras, la tarea de validación de requisitos es realizada de una forma más complicada, ya que se hará uso de diversas técnicas así como también se aumentará la base de la validación con el fin de añadir necesidades y perspectivas de las partes interesadas.

Existen diferentes técnicas que son utilizadas para la realización de la validación de requisitos, y las cuales se mencionan a continuación:

- Análisis
- Simulaciones
- Prototipos
- Demostraciones

Capítulo 10: Sistemas críticos

10.1 Introducción

Una correcta realización de la especificación de requisitos de sistemas críticos, es de gran importancia, ya que, un error en esta, puede tener fuertes consecuencias como provocar la muerte de un ser humano, modificar significativamente la actividad de una empresa u originar un deterioro del medio ambiente. Desde un primer momento se ha de saber que la especificación de requisitos de sistemas críticos es un complemento a la especificación de requisitos software, es decir, no hay opción de elegir entre una u otra.

Por terribles accidentes ocurridos a lo largo de la historia, se ha tomado conciencia de la gran importancia que tiene la especificación de requisitos en sistemas críticos. La principal prioridad en estos casos es garantizar al 100% que la especificación en su conjunto manifiesta exactamente cada una de las necesidades declaradas por los usuarios. En todo momento se ha de asegurar que el sistema es confiable, es por ello que para que se dé esta circunstancia, la especificación de sistemas críticos ha de ser correcta.

Un conjunto de requisitos funcionales y no funcionales son producidos como consecuencia de la confiabilidad en sistemas críticos:

- El conjunto de requisitos funcionales serán creados con el fin de especificar la verificación de errores, habilidad de restauración/recuperación y propiedades que aportan seguridad en el caso de errores o mal funcionamiento del sistema.
- El conjunto de requisitos no funcionales son producidos con el fin de explicar por una parte la fiabilidad y por otra la disponibilidad necesitada por el sistema.
 - *Fiabilidad*: ¿Cuándo un sistema es fiable? Pues bien, un sistema es fiable cuando la forma de actuación de este es razonable, lógica y se corresponde con la especificación.
 - *Disponibilidad*: ¿Para qué debe estar el sistema disponible? Bien, el sistema debe de estar disponible para prestar el servicio para el que ha sido creado tan pronto como sea necesario.

Aparte de dichos requisitos, características como protección y seguridad, aportan otro conjunto de requisitos, los cuales, son difíciles de clasificar en alguno de los grupos anteriores, ya que estos detallan conductas del sistema que de algún modo no son aceptadas. En algunos casos se pueden englobar dentro de los requisitos funcionales, pero estos tienen un alto grado de especificación. Seguidamente nos encontramos con una escueta descripción de lo que se entiende cuando se habla de seguridad y protección:

- *Seguridad*: cuando se habla de que el sistema debe ser seguro, se refiere a que cualquier conducta de este, no debe originar ningún perjuicio al medio ambiente ni a ninguna persona física.
- *Protección*: el sistema y conjunto de datos existentes, deben estar protegidos contra cualquier acceso no autorizado.

A continuación se describen algunos ejemplos de este tipo de requisitos, todos ellos existentes en mi trabajo:

- Requisito de protección: A nuestro sistema pueden acceder un conjunto amplio de agentes que están dados de alta como tales en nuestra compañía y como consecuencia de esto en nuestras bases de datos. Cada agente, dispone o bien de una tarjeta inteligente o bien de un certificado software, con un identificador y una contraseña definidos por la compañía. El identificador está formado por un código numérico, las iniciales del responsable de la tarjeta y el nombre del agente y el pin está formado por un conjunto de caracteres numéricos. Entonces, el sistema no debe autorizar el acceso a aquellas personas o usuarios que no estén dadas de alta como agentes en la BBDD.
- Requisitos de seguridad: mi compañía, como cualquier otra compañía, dispone de un CPD (centro de procesamiento de datos). Este está situado en una sala la cual se encuentra insonorizada con el fin de que el conjunto de operadores que trabajan junto a ella no sufran ningún daño.

Como se sabe, aparte de requisitos funcionales y no funcionales existen otros tipos, entre los que podemos destacar los requisitos de usuario. Señalar que los requisitos de usuario en este tipo de sistemas serán especificados mediante el uso de modelos del sistema y lenguaje natural.

Hasta aquí, se ha hablado de algunas características deseables que han de tener los sistemas críticos, fiabilidad, disponibilidad, seguridad, protección, y aunque no se ha mencionado, también mantenibilidad. Bien, este grupo de características, no son autónomas entre sí, es decir, todas ellas deben cooperar juntas para alcanzar un mismo fin, el funcionamiento correcto y exacto del sistema.

10.2 Especificación dirigida por riesgos

En primer lugar, se empleará la especificación dirigida por riesgos cuando en el sistema, la confiabilidad sea una característica crítica.

En este tipo de especificación, se ha de estudiar y entender el conjunto de riesgos que debe afrontar el sistema, así como también las posibles causas generadoras de estos, con el fin de crear el conjunto de requisitos específicos para tratar estos riesgos de los que hablamos.

10.2.1 Análisis de riesgos

A continuación se describen el conjunto de etapas que han de llevarse a cabo para la realización del análisis de riesgos:

- Identificación de riesgos

Consiste en determinar o señalar el conjunto de posibles riesgos que pueden aparecer en el sistema. Todos ellos dependerán del dominio en el cual el sistema será usado.

La acción de identificar el conjunto de riesgos puede ser complicado, ya que en la mayoría de las ocasiones muchos riesgos surgen como resultado de la interacción entre el sistema y diferentes situaciones casuales del entorno.

- Análisis y clasificación de riesgos

Cada uno de los riesgos obtenidos en la fase anterior será estudiado. Aquellos riesgos en los que la probabilidad de aparición sea muy remota, directamente serán eliminados mientras que aquellos que son de gran importancia y no pronosticables serán analizados posteriormente, por lo tanto, resumiendo, esta etapa consistirá en conocer cuál es la posibilidad de que surja un riesgo así como también cuales son los efectos en el supuesto caso de que se dé un percance relacionado con este.

Existen tres maneras distintas de agrupar los diferentes riesgos, las cuales se mencionan a continuación:

1. Riesgo inaceptable: son aquellos que en caso de darse pueden poner en peligro la vida humana o la estabilidad financiera de una empresa. El porcentaje de probabilidad de aparición de este tipo de riesgos es alto, es por esto que el sistema debe estar protegido para que nunca se den estos, o en el supuesto caso de darse, no causen ninguna desgracia ni ningún percance.
2. Riesgo admisible: aun siendo estos aceptables, los diseñadores persiguen en todo momento minimizar dicha aparición, pero siempre esto entre otros requisitos no funcionales, ha de respetar los costes del proyecto, es decir, no debe aumentarlos y los tiempos de entrega han de ser los mismos, es decir, nunca se deben alargar.
3. Riesgo tan bajo como razonablemente práctico (ALARP): llegados a este punto, los dos tipos de riesgos anteriores, persiguen minimizar siempre la aparición de estos así como también respetar plazos y costos entre otros. Este tercer tipo de riesgos por una parte son aquellos en los que efectos que ocasionan en el caso de darse son menos graves e importantes, y por otra parte la posibilidad de que se den es menor.

A continuación se describe en la siguiente tabla un ejemplo de cada uno de los tipos de riesgos existentes:

	INACEPTABLE	ACEPTABLE	RIESGO TAN BAJO COMO RAZONABLEMENTE PRACTICO
EJEMPLOS	Pérdida total de la base de datos, esta necesaria para el funcionamiento de la empresa	La entrada de un virus a los sistemas, este solo afectando a los Pc's pero sin tener ninguna consecuencia más.	Aunque poco probable o prácticamente imposible, el hecho de que se dé una caída de las comunicaciones y que como consecuencia de esto, los agentes no puedan acceder al sistema durante un corto periodo de tiempo.

Tabla 13: Tipos de riesgos

- Descomposición de Riesgos

En esta etapa, el principal objetivo consiste en revelar cuáles son los principales motivos que dan lugar a cada uno de los riesgos.

El análisis de riesgos puede ser inductivo o deductivo, pero ¿Cuál es la diferencia entre ambos tipos? Bien, en las técnicas inductivas o también llamadas técnicas ascendentes, su punto de partida es una sugerencia de funcionamiento erróneo en el sistema, y como resultado se obtiene un conjunto de posibles riesgos que pueden darse en el supuesto caso de que ocurra dicho error, mientras que las técnicas deductivas o también llamadas técnicas descendentes, parten del riesgo hasta obtener como resultado el probable funcionamiento erróneo del sistema.

Para esto, existen un conjunto de técnicas aplicables, entre las que se puede destacar la técnica del análisis del árbol de defectos. A continuación se describe brevemente en que consiste dicha técnica:

- Técnica del análisis del árbol de defectos: en primer lugar la técnica del análisis del árbol de defectos es una técnica muy sencilla de concebir aun sin poseer ningún tipo de conocimiento. Dicha técnica consiste en reconocer sucesos no esperados y mediante un análisis minucioso de estos, se realiza un retroceso partiendo de dicho suceso con el fin de revelar los factibles motivos que dan lugar al riesgo. Para ello se debe situar el riesgo como raíz del árbol y reconocer el conjunto de estados que pueden dar lugar a este riesgo. El siguiente paso, para cada uno de los estados identificados, se deben reconocer el conjunto de estados que dan lugar a este y así sucesivamente, con el fin de llegar a conocer el principal motivo que da lugar al riesgo. Como nota aclaratoria, indicar que el conjunto de estados pueden ser enlazados a través de los símbolos “or” (ó) y “and” (y).

A continuación se incluye un ejemplo relacionado con mi empresa para que esta técnica se comprenda mejor:

“En mi compañía es imposible quedarnos sin comunicaciones debido a que si el sistema eléctrico falla, entre otros mecanismos se dispone de un grupo electrógeno que lo que hace es suministrar energía durante un número muy elevado de horas. Este grupo electrógeno se prueba todos los meses, para ello se provoca un simulacro mediante un corte eléctrico, por lo que dicho grupo debe de arrancar y funcionar hasta que la electricidad vuelva, todo esto se hace con el fin de verificar que el funcionamiento es el correcto.” Partimos del punto que si el grupo electrógeno no funciona, es un suceso no esperado.

Nos ponemos en el caso, de que un día cualquiera, se da un corte de la electricidad, y el grupo electrógeno no arranca, cuando este se ha probado rutinariamente todos los meses y no ha habido ningún problema.

Entonces, llevando a cabo la técnica del análisis del árbol de defectos, tenemos que revelar los factibles motivos que pueden dar lugar a esto (solo señalaremos un motivo para no alargar esto):

- El grupo electrógeno que funciona con gasoil, y debe tener una cantidad mínima necesaria, que no disponga de la suficiente.

Ahora, para este estado anterior, debemos de reconocer lo que puede llevar a esto:

- Que el técnico de mantenimiento que es el encargado de supervisar los niveles de gasoil, no los haya revisado después de las pruebas y por consiguiente el grupo no disponga de gasoil suficiente para arrancar.
- Que el grupo electrógeno tenga una fuga, es decir que se haya averiado y el técnico de mantenimiento no se haya percatado de ello.

Así se seguiría analizando, hasta finalmente encontrar el motivo principal que ha dado lugar a la incidencia.

- Valoración de la reducción de riesgos

Llegados a este punto se realizan un conjunto de sugerencias, las cuales engloban los posibles modos o maneras en las que cada riesgo puede ser disminuido o suprimido directamente. Como consecuencia de esto, se crean un conjunto de requisitos de confiabilidad que especifican los métodos de protección frente a los riesgos, así como también la forma en la que serán tratados estos en caso que se produzcan. El objetivo principal de los requisitos de confiabilidad consiste en prevenir que se den incidentes y contratiempos/accidentes.

- *Accidente*: sucesión programada de acontecimientos que culmina con la muerte de un ser humano o con deterioros en el entorno de los sistemas.
- *Incidente*: cualquier situación que altere la situación normal de los sistemas.

Existen tres tipos de tácticas que pueden ser seguidas en la fase en la que nos encontramos:

1. *Eludir riesgos*: el sistema es creado para evitar que puedan existir riesgos.
2. *Interrupción y anulación de riesgos*: el sistema es creado con el fin de que los riesgos sean localizados y contrarrestados antes de que estos causen un incidente.
3. *Restricción del perjuicio*: el sistema es creado para que los efectos de un accidente sean los menos posibles.

En cuanto al uso de estos, indicar que generalmente lo que se suele hacer es usar una combinación de todos ellos.

Capítulo 11: Validación de requisitos

11.1 ¿En qué consiste la fase de validación de requisitos?

Recordamos que el principal objetivo del proceso de validación de requisitos consiste en comprobar el conjunto de requisitos, es por ello, que en dicha verificación se pueden encontrar algunos problemas relacionados con los requisitos como por ejemplo que estos no estén bien redactados y dé lugar a ambigüedad, que no cumplan adecuadamente los estándares de calidad o que exista conflicto entre requisitos no detectados en etapas tempranas.

El conjunto de problemas encontrados en la fase de validación, deben ser resueltos antes de que el documento de especificación de requisitos sea aceptado, es por esto que como solución a dichos problemas se ha de retroceder a las etapas anteriores (obtención, análisis y negociación de requisitos) con el fin de solucionarlos.

11.2 Técnicas relacionadas con la validación de requisitos

A continuación se mencionan un conjunto de técnicas existentes para realizar la validación de los requisitos, las cuales pueden ser utilizadas de forma individual o como combinación de algunas de ellas.

Respecto al conjunto de técnicas existentes para la validación de requisitos, hay que señalar que las técnicas más habituales y utilizadas son elaboración de prototipos, verificación de requisitos y creación de casos de pruebas, mientras que la realización de manuales de usuario y animación y validación de modelos, son técnicas alternativas.

11.2.1 Técnica basada en la elaboración de prototipos

Cuando se utiliza este método, el objetivo es crear un modelo idéntico al sistema real (maqueta), con el fin de que usuarios finales prueben y examinen si la funcionalidad de dicho prototipo cumple con las necesidades y exigencias.

Cuando nos decantamos por el uso de un prototipo, se tienen dos partes claramente diferenciadas, por una parte se tiene la construcción de este, que es la más sencilla, y por otra un conjunto de tareas que engloban todas las actividades restantes.

Existen diferentes tipos de prototipos, por lo que dependiendo del tipo que sea utilizado, tanto la clase de pruebas a poder realizar como el nivel de realismo dependerán de esto. A continuación se mencionan tres tipos de prototipos, los más utilizados en ingeniería de requisitos:

- Mock-ups: consiste en dibujar a mano un conjunto de pantallas con el fin de que estas representen la apariencia del sistema. Este tipo de prototipos no aportan gran información a la validación, pero sí aclara la interfaz gráfica en hechos complejos.
- Storyboards: son un paso más a los mock-ups, ya que aparte de mostrar la interfaz gráfica, también aporta información acerca del encadenamiento de acciones que deben ser realizadas mediante el programa.

- Maquetas: el principal objetivo de una maqueta es representar la interfaz del sistema y en ciertas ocasiones las relaciones entre pantallas mediante el uso de botones. Normalmente, las maquetas se realizan mediante Visual Basic o PowerBuilder.

Una vez que nos encontramos en el punto de tener que construir el prototipo, ya se conocen los diferentes tipos de prototipos que existen, pero probablemente nos surja la duda de cuál utilizar en cada ocasión, pues bien, el utilizar uno u otro dependerá de los factores de los que se disponga. A continuación se muestra un cuadro-resumen donde esto se aclara:

FACTORES DISPONIBLES	PROTOTIPO RECOMENDADO	¿POR QUE UTILIZAR EL PROTOTIPO INDICADO?
<ul style="list-style-type: none">- Recursos disponibles- Tiempo disponible	Mock-ups Storyboards	Cuanto más limitados sean los recursos, más recomendable es construir prototipos sencillos.
<ul style="list-style-type: none">- Fidelidad deseada	Prototipos complejos	El hecho de que el usuario no comparte la idea de utilizar prototipos, debido a que estos no son capaces de comprender que un prototipo es una versión minimizada del sistema final, cuanto más complejos sean mejor. Además el analista debe encargarse de informar de todo aquello que el prototipo no pueda proveer.

Tabla 14: Relación entre los factores disponibles y el prototipo a utilizar

Como se ha mencionado en párrafos anteriores, aparte de la construcción del prototipo también existe la necesidad de realizar un conjunto de tareas, las cuales se detallan a continuación:

- ❖ Elección de personas encargadas de evaluar el prototipo: en la evaluación del prototipo participarán un conjunto de personas, a ser posible con perfiles diferentes, con el fin de que dicha actividad sea efectiva. Se pone gran énfasis en que tengan diferentes perfiles, ya que de esta manera el software será validado en todos sus modos de utilización.
- ❖ Creación de escenarios con el fin de validar el prototipo: para la realización de una evaluación correcta del prototipo, se ha de realizar una batería de escenarios, con la intención de que estos sean probados por los diferentes usuarios empleando el prototipo.

- ❖ Realización de escenarios: en esta fase, el conjunto de escenarios desarrollados en la etapa anterior, deben ser probados por los usuarios. En principio se sabe que en dichas pruebas, existirán diferencias entre el prototipo y el producto final, es por ello que el analista debe encargarse de informar al usuario de dichas diferencias, con el fin de que este no se manifieste disconforme. Además, el analista también debe proporcionar la información que el usuario necesite para la realización de los escenarios y que el prototipo no ofrezca.
- ❖ Confección de listado de problemas: consiste en documentar los problemas encontrados. Ya que el analista encargado de la especificación de requisitos es el mismo que el que participa en la validación junto a los diferentes usuarios, no es común realizar una lista de recomendaciones.

En cuanto a las correcciones que deben ser realizadas por los problemas encontrados, existen dos posibles caminos a seguir. Una primera opción consiste en ir subsanando los problemas encontrados con el fin de ir mejorando el prototipo, y una segunda opción consiste en terminar completamente la validación y dejar todas las correcciones para el final. El hecho de decantarse por una opción u otra, dependerá del tipo de anomalías encontradas.

11.2.2 Técnica basada en la verificación de requisitos

La verificación de requisitos consiste en sucesivas reuniones e implica a un conjunto amplio de personas, cuyo principal objetivo es comprobar que en el documento de especificación de requisitos no existen anomalías. Dicho método permite minimizar los costes de desarrollo, revelar errores en los requisitos, estos con un porcentaje muy elevado, que radica entre el 50-90% y minimizar el tiempo de pruebas.

Existen dos tipos de revisiones, las revisiones formales y revisiones informales, donde cada una de ellas tiene las siguientes características:

- ✓ Revisiones formales: en este tipo de reuniones, el cliente debe ser arrastrado por el equipo de desarrollo hacia los requisitos definidos del sistema, revelándole en cada caso las implicaciones de cada uno de ellos.
- ✓ Revisiones informales: en este tipo de reuniones, el objetivo es que el contratista analice los requisitos con el mayor número posible de stakeholders.

Las sucesivas reuniones que han de ser realizadas para llevar a cabo la técnica de verificación de requisitos, están compuestas por un conjunto de pasos, en concreto siete, los cuales se describen a continuación:

Paso 1: Organizar un guion de la revisión

Contiene las tareas que han de ser realizadas, los recursos humanos implicados, es decir, las personas que deben colaborar en la revisión así como también el cronograma temporal para la realización.

Paso 2: Reparto de la documentación a revisar

Análisis de requisitos en el desarrollo del software

Como ya se sabe, el documento a revisar en el proceso de validación de requisitos es el documento de especificación de requisitos software, pero en caso de existir, también hay que adjuntar anexos y documentos relacionados con la especificación. Todo este conjunto de documentos e información deben ser distribuidos entre los implicados en esta actividad.

Paso 3: Organizar la reunión

Esta tarea puede ser realizada o bien por los analistas colaboradores o bien por el analista organizador de la reunión, bien, pues dependiendo de quién realice dicha actividad esta será enfocada de diferente manera.

- Cuando esta tarea es realizada por los *analistas colaboradores*, estos leerán metódicamente la documentación disponible y tomarán notas de aquellos aspectos que ellos no consideran correctos, con el fin de discutirlos en la próxima reunión.
- Cuando dicha tarea es realizada por el *analista organizador* de la reunión, este básicamente se encargará de localizar y reservar una sala donde llevar a cabo la reunión, así como también será el encargado de demandar los materiales necesarios, refiriéndonos con esto a cuartillas, bolígrafos y proyectores entre otros.

Paso 4: Llevar a cabo reunión de revisión

Paso 5: Detallar anomalías y posibles acciones a seguir

El conjunto de anomalías y de acciones aconsejables a realizar, es la salida que se obtiene en esta fase de validación de requisitos, es decir, el objetivo una vez finalizada la revisión consiste en poseer un documento donde se detalle para cada defecto la información referente al número de requisito, al error detectado en dicho requisito y el conjunto de acciones recomendadas a llevar a cabo para solucionar dicha anomalía.

Numero de requisito	Error detectado	Acciones recomendadas
7	Ambigüedad	Redactar en el lenguaje correcto, de manera que no de opción a distintas interpretaciones.

Tabla 15: Detalle de anomalía y posibles acciones a tomar para solventarla

Paso 6: Realizar las modificaciones necesarias en los documentos chequeados

Una vez que ya se dispone del listado de anomalías encontradas y las posibles acciones a llevar a cabo, el analista organizador de la reunión será el encargado de analizar dichas propuestas realizadas para solventar los problemas, y en caso de considerarlas aptas, llevarlas a cabo.

Paso 7: Notificar a los colaboradores de la reunión los cambios realizados

Una vez que las modificaciones que han sido consideradas viables han sido realizadas, lo correcto es enviar al conjunto de colaboradores de la reunión, por una parte un informe con el conjunto de actividades realizadas y por otra parte una copia del documento de especificación de

requisitos software con los cambios ya realizados, es decir, el documento final, con el fin de que el conjunto de colaboradores den el visto bueno.

11.2.3 Técnica basada en la creación de casos de pruebas

Cada uno de los requisitos debe de poder ser probado, por lo que en cada caso de prueba se comprobarán diferentes funcionalidades. Se ha de tener en cuenta, que si un requisito es complicado de probar mediante un caso de prueba, entonces significa que dicho requisito será difícil de implementar.

Por otra parte, cada uno de los requisitos debe cumplir unos estándares de calidad, bien, pues uno de ellos consiste en que los requisitos sean verificables. Esto es debido a dos razones:

- ❖ Primera razón: el conjunto de requisitos debe ser verificable para que se pueda comprobar el cumplimiento o no cumplimiento de este durante las pruebas.
- ❖ Segunda razón: este atributo de calidad requiere que cada uno de los requisitos sean estables, íntegros y estén perfectamente enunciados.

La utilización de este método, genera dos resultados:

- ❖ Listado de anomalías encontradas con acciones recomendadas a seguir para subsanar estos.
- ❖ Colección de posibles casos de pruebas que pueden ser usados en las siguientes pruebas del sistema.

11.2.4 Animación y realización de modelos

El principal objetivo de esta técnica consiste en comprobar que el funcionamiento del sistema es el esperado por los diferentes usuarios, así como también que actúa de la misma manera a cómo está especificado.

11.2.5 Realización de manuales de usuarios

Este método de validación de requisitos es muy parecido al método de creación de casos de prueba, donde se persigue comprobar si la información que engloba el documento de especificación de requisitos tiene un alto grado de detalle como para realizar el manual de usuario del sistema.

11.3 Observaciones de interés en la fase de validación de requisitos

Llegados a este punto, si nos paramos a pensar, parece que los procesos de análisis y validación de requisitos, por las tareas que engloban cada uno de ellos, son actividades muy similares, ya que en ambos los propósitos que se persiguen son muy iguales, como por ejemplo que no haya conflicto entre requisitos o que no haya ambigüedad, pero según la descripción que se va a dar a continuación, se observará como esto no es del todo cierto:

- ❖ Respecto a la actividad de análisis, el conjunto de requisitos que se analizan provienen como ya se conoce de un amplio grupo de stakeholders. En la mayoría de las ocasiones estos requisitos suelen ser incompletos y están expresados de un modo no formal.
- ❖ Sin embargo, en la validación de requisitos, la verificación ya es realizada sobre un documento de especificación de requisitos formal, donde cada uno de ellos es completo y consistente, además de que dicho documento contiene todos los requisitos del sistema.

En la validación de requisitos nos encontramos con un problema, ya que no se puede validar el documento de especificación de requisitos frente a nada, ya que de lo único que se dispone es de dicho documento. Luego la validación en cierto modo lo que aporta es seguridad, ya que se revisa que el conjunto de requisitos son aptos para su posterior implementación y desarrollo, además de que esta es la última oportunidad para realizar modificaciones antes de entrar en desarrollo el sistema.

Para la actividad de validación de requisitos hay que emplear el tiempo necesario, ya que se parte de la base de que es una actividad larga, luego no sirve con precipitar su curso natural, ya que en el caso de no emplear el tiempo suficiente, la probabilidad de que los requisitos no sean correctos es demasiado alta. No obstante, se puede empezar con el desarrollo del sistema de forma precipitada, pero hay que estar preparado para el posible retroceso a etapas anteriores, ya que es muy probable que los requisitos sean erróneos.

Como se lleva hablando durante todo este capítulo, en esta fase, el documento de especificación de requisitos debe ser revisado con el fin de verificar el conjunto de estos. Bien, esta revisión será realizada por un grupo de personas, pero antes de efectuarse el reparto de dicho documento entre los implicados, ha de realizarse una revisión general para confirmar que el documento en general, y los requisitos en particular, son estables y cumplen los estándares. Este trabajo será llevado a cabo por una única persona, lo que implica una serie de beneficios como por ejemplo que dicha revisión es muy económica, ya que es realizada por una única persona. Razonando, es comprensible que esta sea llevada a cabo por una única persona, ya que de no ser así, implicaría a más personas y por ello sería mucho más costoso, y además los resultados serían los mismos, ya que al final todos los involucrados llegarían a hacer las mismas aportaciones.

La persona encargada de revisar el documento, debe tener conocimiento previo acerca de los estándares, pero sin embargo, no debe haber estado involucrada en el proceso de realización del documento de especificación de requisitos. Dicha persona será la encargada de realizar una comparación entre el documento de especificación de requisitos y la norma o estándar existente, revisando que el documento y todas las secciones están completos.

Para realizar la validación de requisitos, es necesario disponer de un listado de verificaciones (checklist), previamente definido, con el fin de que las personas encargadas de realizar dicha actividad reconozcan cuales son los puntos que deben validar.

El disponer de dicho listado, por una parte al conjunto de personas que son nuevas en el proceso de validación de requisitos, les ayuda, ya que este proporciona pistas para que puedan realizar el proceso y por otra parte dicho listado de verificaciones agrega una estructura, por lo que

es más complicado que los usuarios encargados de realizar dicha actividad omitan comprobar detalles del documento de especificación de requisitos software.

El listado englobará un conjunto de cuestiones, cuyo principal objetivo es ayudar al analista a reconocer las anomalías existentes en el documento de especificación de requisitos. Es recomendable que cada empresa disponga de su propia lista de verificaciones acondicionada al tipo de sistemas que desarrolla, ya que de esta manera siempre se obtendrán mejores resultados que si se utiliza un listado de verificaciones genérico.

A continuación se listan algunos ejemplos de preguntas que podrían ser incluidas en la lista de verificaciones:

- ¿Existe ambigüedad en los requisitos? – Comprobar que cada uno de los requisitos solo tiene una interpretación, no da lugar a diferentes interpretaciones.
- ¿El documento de especificación de requisitos software cumple con los estándares?
- ¿Son el conjunto de requisitos coherentes? – Comprobar que los requisitos no se contradicen entre ellos.
- ¿El conjunto de requisitos son comprensibles? – Comprobar que las personas que utilizan el documento lo comprenden sin problema.

El conjunto de preguntas que se engloban dentro del listado de verificaciones, deben estar definidas de tal manera que estas sean comprendidas por todos los usuarios, partiendo de la base de que muchos de ellos no tienen ningún conocimiento previo del sistema, como por ejemplo los usuarios finales, y además dicho listado debe ser escueto, es decir, la longitud de este no debe superar los diez puntos.

Finalizada la revisión y una vez que ya tenemos los resultados de esta, existen dos posibles alternativas a seguir:

Alternativa 1

Encargarse el grupo de ingeniería de requisitos de modificar los errores encontrados, siempre y cuando se disponga de tiempo suficiente para ello, ya que el hecho de que el documento vuelva a este grupo implica una re-edición del documento y la realización de todos los pasos que esto conlleva.

Alternativa 2

Encargarse el grupo de personas que han participado en la revisión. ¿Cuándo realizará las modificaciones este grupo? Cuando el tiempo del que se dispone es pequeño y además las modificaciones que han de ser realizadas son simples y no perjudican a la comprensibilidad del documento. Además, el hecho de que sean realizadas por este grupo, significa por una parte un ahorro de tiempo, ya que el documento no pasa a los ingenieros de requisitos y de nuevo debe ser revisado y por otra parte un ahorro de costos.

Resumiendo, de toda la información que ha sido aportada acerca de la validación de requisitos, llegados a este punto, se debe conocer que como entradas al proceso de validación se tienen el documento de especificación de requisitos, los estándares y normas ya existentes y por último el conocimiento previo de la organización y como salida se obtiene un listado de problemas

encontrados y un listado de acciones a realizar. A continuación se muestra dicha información gráficamente:



Ilustración 19: Proceso de validación de requisitos [1]

[1] Tomado de http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf

Capítulo 12: Calidad de proyectos software

12.1 Información general sobre calidad

12.1.1 Definición de calidad

En esta sección en la que nos encontramos, nos centraremos especialmente en calidad y en todos los aspectos relacionados con esta, ya que hoy en día, la calidad ha pasado a ser uno de los primordiales objetivos estratégicos para las organizaciones, ya que, en cierto modo, la competitividad de estas depende de la calidad de productos y servicios que ofrece ofrecen a usuarios y clientes.

A continuación se proporciona una definición de “calidad”:

- ❖ Según el diccionario de la Real Academia Española de la Lengua (DRAE, Oct. 2014), el término calidad, en sus cinco primeras acepciones, es definido como:
 1. Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.
 2. Buena calidad, superioridad o excelencia.
 3. Carácter, genio, índole.
 4. Condición o requisito que se pone en un contrato.
 5. Estado de una persona, naturaleza, edad y demás circunstancias y condiciones que se requieren para un cargo o dignidad.

Además, en algunas normas internacionales existentes hoy en día, como por ejemplo en la **ISO 8402**, también podemos encontrar la definición del término de calidad, el cual se define como:

“Conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas”.

12.1.2 Tipos de orígenes de la calidad

Cabe mencionar, que la calidad puede tener diferentes orígenes, probablemente nos preguntemos ¿Qué significa esto?, bien, pues a continuación se realiza una descripción detallada:

<u>ORIGEN</u>	<u>DEFINICION</u>
<u>Calidad Realizada</u>	Tiene que ver con la calidad que es capaz de conseguir el individuo encargado de hacer el trabajo, todo ello gracias a la capacidad para la realización de una tarea. Esta capacidad de la que se está hablando, se incrementa con el perfeccionamiento de las habilidades personales de los colaboradores de un proceso.
<u>Calidad programada</u>	Se refiere a la calidad que se ha perseguido obtener, es decir, es la que figura en un documento de especificación, y es por ello, que es la que se le ha encargado de alcanzar al responsable de llevar a cabo el

	trabajo.
<u>Calidad Necesaria</u>	Se refiere a la calidad que el cliente demanda, es decir, la calidad que el cliente quiere recibir. Esto, se incrementa con una correcta consecución de información acerca de la calidad de los clientes.

Tabla 16: Definiciones de los diferentes tipos de calidad

Destacar que en todo momento se persigue que los tres tipos de orígenes anteriores coincidan lo más posible. Además, siempre se analiza si la calidad esperada por el cliente coincide con la calidad realizada, ya que un porcentaje elevado de los problemas existentes de la calidad tienen origen en las falsas expectativas provenientes del cliente sobre las propiedades de un servicio o producto.

12.1.3 Tipos de vistas relacionadas con calidad

Dependiendo de la persona que evalúe el cumplimiento de la calidad, según [Gar84], existen cinco “vistas”, las cuales se describen a continuación:

<u>TIPO DE VISTA</u>	<u>DEFINICION</u>
<u>Vista basada en el valor</u>	La calidad depende de la cuantía de dinero que el cliente quiera pagar.
<u>Vista de fabricante</u>	Según este tipo de visión, la calidad está centrada principalmente en el proceso, y mantiene que esta es de acuerdo a las especificaciones, por lo que su alcance se extiende con el fin de comprobar también la calidad durante la producción del producto y posteriormente a la entrega de este.
<u>Vista trascendental</u>	La calidad es algo que se intenta perseguir alcanzar, que nunca se llega a conseguir debido a diferentes factores como por ejemplo la tecnología y que no se define como tal.
<u>Vista del producto</u>	Se parte del punto de que la calidad y el conjunto de características esenciales del producto están unidos. Sin embargo las vistas del fabricante y usuario se obtienen “desde fuera”, por el contrario de las del producto, que se tienen “desde dentro”, debido a que estas últimas están relacionadas con la medida de las propiedades internas de los productos.
<u>Vista de usuario</u>	La calidad siempre es adaptación al propósito, es por ello que se puede expresar numéricamente cada una de las características del producto, y por consiguiente determinar los

	propósitos a alcanzar.
--	------------------------

Tabla 17: Definición de los tipos de vistastomadas de [Gar84]

12.1.4 Actividades que engloba la gestión de calidad

La manera de gestionar la calidad dependerá en gran medida del tamaño del proyecto. Esto es, cuando nos encontramos frente a un proyecto de gran tamaño y complejidad, es de gran importancia realizar una gestión de calidad de una manera formal, refiriéndonos con esto a la documentación a cumplimentar por cada subgrupo del proyecto. Sin embargo, si nos encontramos frente a un proyecto de pequeño tamaño, la gestión de calidad tiene la misma importancia que en el caso anterior, pero siempre es de manera más informal.

¿Por qué existe esta diferencia de trato dependiendo del tamaño del sistema? Esto es, porque en un proyecto de gran tamaño, la documentación pasa por un gran número de personas, por lo que estos tienen que estar completamente actualizados de los cambios que los demás miembros han hecho, mientras que en proyectos de menor tamaño, al ser el número de personas igualmente reducido, simplemente basta con asegurarse de que todos los integrantes del proyecto están informados de las modificaciones.

La gestión de la calidad como tal, engloba tres actividades, las cuales se indican a continuación:

<u>ACTIVIDAD</u>	<u>DEFINICION DE CADA ACTIVIDAD</u>
Garantía de la calidad	Esta primera actividad consiste en establecer un marco de trabajo de estándares organizacionales y procedimientos, cuyo objetivo es la producción de software de alta calidad.
Planificación de la calidad	Elección de los procedimientos y estándares apropiados, así como también adecuar estos a un determinado proyecto software.
Control de la calidad	Definición de procesos específicos que aseguren que los estándares y procedimientos anteriores son llevados a cabo por el equipo de desarrollo.

Tabla 18: Actividades que forman la gestión de calidad

Debido a que dentro de calidad es un tema que me parece de gran importancia, a continuación, en los puntos anteriores se habla más en profundidad de la planificación, control y revisión de calidad.

12.1.4.1 Planificación de la calidad

La planificación de la calidad es la primera etapa, donde el principal objetivo es crear un plan de calidad para un proyecto específico. Generalmente, un plan de calidad por una parte define la

calidad que nos gustaría que tuviera el software, y por otra especifica la manera de valorar esta. Esto es, se especifican los atributos de calidad de mayor importancia para el producto a desarrollar, aunque la eficacia es uno de los que mayor prioridad tiene.

Existen distintas sugerencias acerca de la organización de un plan de calidad, según [Humphrey, 1989], este engloba lo siguiente:

Punto 1: Introducción del producto

Este punto engloba información acerca del producto, así como también de las expectativas de calidad y del mercado al que va dirigido.

Punto 2: Planes de producto

Los planes de producto lo forman las fechas de finalización del producto, responsabilidades de gran importancia y por último los planes para el servicio y distribución.

Punto 3: Descripciones del proceso

Informa acerca de los procesos de desarrollo y servicio que es necesario usar para el desarrollo y control del producto.

Punto 4: Metas de calidad

Contiene información acerca de objetivos y planes de calidad para el producto en cuestión, englobando en ello la identificación y argumento de los atributos de mayor importancia de calidad del producto.

Punto 5: Riesgos y gestión de riesgos

Este punto aporta información acerca de los riesgos importantes que pueden llegar a perjudicar a la calidad así como también las acciones a llevar a cabo en caso de darse ese tipo de situaciones.

Evidentemente, igual que ocurre con el documento de especificación de requisitos software, el plan de calidad dependerá de diferentes factores relacionados con el sistema a desarrollar, como por ejemplo el tamaño de este, pero aunque el tamaño es un factor que influye, siempre independiente de esto hay que intentar que el plan de calidad no sea muy extenso, ya que de no ser así los ingenieros acabarían por no leerlo.

12.1.4.2 Control de la calidad

Como su nombre indica, el principal objetivo del control de calidad consiste en realizar un seguimiento con el fin de verificar que se siguen los estándares de garantía de calidad así como también los procedimientos. Además es de vital importancia verificar en esta fase que las distintas entregas cumplen los estándares detallados.

En un proyecto, con el fin de revisar las entregas de este, hay dos tipos de enfoques complementarios:

1. Se realiza una revisión generalizada sobre aspectos como el software, los procesos usados y documentación existente, todo esto llevado a cabo por un grupo de personas. Además, dicho grupo además se ocupa de verificar que el conjunto de aspectos arriba indicado se ajusta a los estándares. En caso de encontrar anomalías o desviaciones, deben ser comunicadas al gestor del proyecto.
2. Este otro enfoque consiste en que los documentos y el software se procesan por específicos programas destinados a ello, donde lo que hacen estos es realizar una comparación entre lo anterior y los estándares que se manejan en el proyecto. Como resultado a esto se obtendrá un valor cuantitativo en relación a algunos atributos del software.

12.1.4.2.1 Revisiones relacionadas con calidad

Hay que tener en cuenta que las revisiones de calidad son la herramienta más usada para verificar la calidad de un producto. Dichas revisiones serán llevadas a cabo por un grupo de personas donde principalmente estudiarán como se ha indicado en el apartado anterior el software, el conjunto de documentos asociados a los sistemas con el propósito de descubrir posibles problemas potenciales.

El grupo encargado de realizar las revisiones de la calidad rutinaria estará formado normalmente por tres o cuatro personas, en el que uno de ellos tendrá el rol de “*diseñador principal*” y cuya función principal consistirá en tomar decisiones técnicas.

El tiempo empleado para la revisión de la calidad es relativamente corto y suele durar dos horas a lo sumo. Además en muchos casos, la documentación es distribuida anterior a la revisión, con la finalidad de que los revisores posean de suficiente tiempo para comprenderla, ya que es de vital importancia que sea comprendida, ya que de no ser así, la revisión no sería eficaz.

En modo informativo y para que la información aportada acerca de la revisión de calidad sea más amplia, después se adjunta una tabla con los tipos de revisiones que existen:

<u>TIPO REVISION</u>	<u>OBJETIVO PRINCIPAL</u>
Inspecciones de diseño	Este tipo de revisión lo que hace es que revisa el código, requisitos y el diseño con el principal objetivo de dar con errores finos. Esta revisión será llevada a cabo mediante una lista de chequeos de los probables errores.
Revisiones de calidad	Consiste en aportar información acerca del avance del proyecto, esta de utilidad para la gestión de este. En este tipo de revisión se chequea tanto el producto como el proceso en temas de planificación, costes y duración.
Revisiones del progreso	Consiste en realizar un análisis técnico de la documentación o en muchos casos de los componentes con el fin de hallar diferencias entre la especificación y diseño de componentes, documentación o código y también con el fin de cerciorar que se siguen los estándares de calidad

	establecidos.
--	---------------

Tabla 19: Tipos de revisiones

12.2 Herramientas y técnicas de calidad

Las herramientas de calidad las forman un conjunto de técnicas y métodos, cuyo principal objetivo consiste en recopilar información con el fin de desarrollar un escenario de calidad.

Existen diferentes tipos de herramientas y técnicas de calidad, así como también diferentes tipos de clasificaciones de estas, pero en nuestro caso nos basaremos en la clasificación de Okes (2002). Debido a que estamos hablando de un número elevado de estas, a continuación se muestra un cuadro-resumen de todas ellas, siendo estas posteriormente explicadas.

1. <i>HERRAMIENTAS BASICAS DE CALIDAD</i>	1.1 Diagrama de flujo
	1.2 Diagrama causa-efecto
	1.3 Diagrama de Pareto
	1.4 Hoja de comprobación
	1.5 Diagrama de control
	1.6 Histograma
	1.7 Diagrama de Correlación
2. <i>HERRAMIENTAS DE GESTION</i>	2.1 Diagrama de afinidad
	2.2 Diagrama de relaciones
	2.3 Diagrama matricial
	2.4 Matriz de análisis de datos
	2.5 Diagrama de redes de actividades
	2.6 Diagrama de árbol
	2.7 Diagrama de proceso de decisiones
3. <i>HERRAMIENTAS</i>	

DE CREATIVIDAD	-----
4.	4.1 Control estadístico del proceso
HERRAMIENTAS ESTADISTICAS	4.2 Diseño de experimento
5.	5.1 Quality Function Deployment (QFD)
HERRAMIENTAS DE DISEÑO	5.2 Análisis Modal de Fallos y Efectos (AMFE)
6.	6.1 Coste de la calidad (COQ)
HERRAMIENTAS DE MEDICION	6.2 Benchmarking
	6.4 Encuestas

Tabla 20: Herramientas y técnicas de calidad

A continuación se proporciona una explicación rápida de cada una de las técnicas arriba mencionadas.

12.2.1 Herramientas básicas de calidad

12.2.1.1 Diagrama de flujo

¿Cuál es el principal objetivo cuando se utilizan los diagramas de flujo?

La principal finalidad de estos es descomponer los diferentes pasos de un proceso en una secuencia. Se pueden utilizar elementos como materiales que salen o entran en el proceso, las personas que participan y secuencia de acciones entre otros.

¿Cuándo se deben utilizar los diagramas de flujo?

Estos deben ser utilizados siempre que se trate de describir el desarrollo de un proceso o bien cuando se intenta establecer una comunicación entre individuos vinculados con el proyecto.

¿Qué simbología se suele utilizar?

Algunos de los símbolos que se utilizan cuando se usan diagramas de flujos son los siguientes:


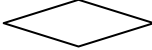
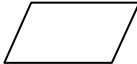

SIMBOLO	SIGNIFICADO
	PROCESO
	DECISION
	DATOS
	TERMINADOR

Tabla 21: Símbolos utilizados en diagramas de flujos

¿Cuáles son los pasos recomendados para realizar un diagrama de flujo?

Para desarrollar un diagrama de flujo, es recomendable seguir cuatro pasos claramente diferenciados, y que son los siguientes:

1. Dar a conocer el proceso que va a ser representado.
2. Reconocer y explicar las distintas actividades que deben ser desarrolladas así como también el orden de estas.
3. Mediante el uso de cajas y flechas representar las actividades y las transiciones, de tal forma que sea posible realizar una traza de este desarrollo.
4. Comprobar el diagrama de flujo resultante con otros individuos para verificar que este es correcto.

12.2.1.2 Diagrama causa-efecto

¿Cuál es el principal objetivo cuando se utilizan los diagramas causa-efecto?

Los diagramas causa-efecto son una herramienta que principalmente se usa para inspeccionar, reconocer y exponer el conjunto de las posibles razones (causa) de un problema determinado (efecto).

¿Qué simbología se suele utilizar?

En este tipo de diagramas únicamente se utilizan dos tipos de símbolos, por una parte un rectángulo, el cual se corresponde con el efecto que se quiere estudiar, y este debe estar situado en

la parte derecha del diagrama, y por otra parte un conjunto de flechas, una central con dirección al rectángulo anterior, y por otra parte otro conjunto de flechas cuyo destino es esta flecha principal y las cuales contienen información acerca de los posibles focos.

¿Cuáles son los pasos recomendados para realizar un diagrama de flujo?

Para desarrollar un diagrama causa-efecto, es recomendable seguir seis pasos claramente diferenciados, y que son los siguientes:

1. Preparar un enunciado claro con el problema descubierto (efecto).
2. Representar el rectángulo con el efecto, así como también la flecha central.
3. Reconocer aproximadamente entre tres y seis flechas secundarias con las principales causas del problema. Dibujar estas con dirección a la flecha principal.
4. Reconocer causas de primer nivel conectadas con la flecha mayor.
5. Reconocer causas del segundo nivel relacionadas con las flechas del primer nivel y así sucesivamente.
6. Analizar los resultados obtenidos y establecer la causa raíz que aporte resoluciones en la conclusión del problema.

A continuación se incluye un ejemplo relacionado con mi trabajo:

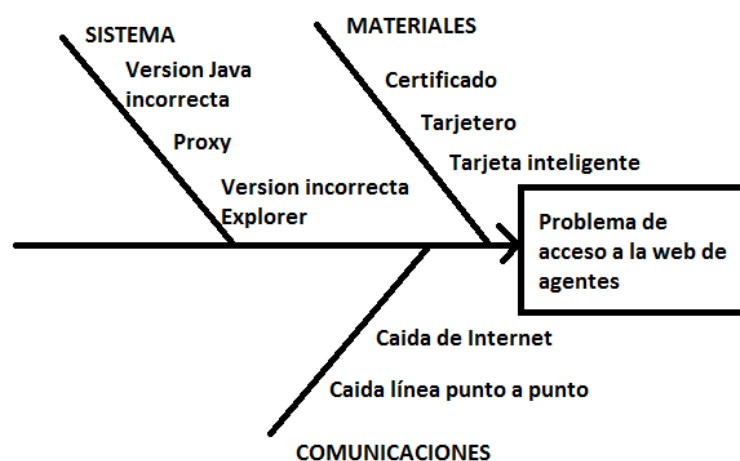


Ilustración 20: Ejemplo diagrama de flujo sobre los problemas de acceso al web de agentes

En dicho diagrama el efecto a analizar es el problema de acceso a la web de agentes, señalar que he decidido poner el ejemplo con esto, ya que en la actualidad es uno de los mayores problemas que tienen los agentes. Se puede observar que las causas principales pueden ser tres, por una parte el sistema, por otra los materiales y por último las comunicaciones. Dentro de cada una de estas, las causas pueden ser varias, por ejemplo por explicar alguna de ellas, cuando se tiene problema con los materiales, puede ser por tres causas conocidas, bien porque la tarjeta inteligente o el certificado esté caducado o bloqueado, o bien porque el tarjetero no funcione correctamente.

12.2.1.3 Diagrama de Pareto

¿Cuál es el principal objetivo cuando se utilizan los diagramas de Pareto?

Dichos diagramas son una herramienta que se usa para establecer una jerarquía del conjunto de problemas o de las razones que los generan mediante una representación gráfica de los datos conseguidos. El principal objetivo de esto es aportar ideas acerca del orden en el que deben ser abordados dichos problemas.

¿Cuáles son los pasos recomendados para realizar un diagrama de Pareto?

Bien, para realizar este tipo de diagramas, es recomendable seguir los nueve siguientes pasos:

1. Reconocer el problema que se quiere analizar
2. Crear un documento donde se almacenen el conjunto de datos recopilados acerca de las causas a inspeccionar así como también la cantidad de veces que aparecen.
3. Juntar todos los datos disponibles y realizar el cálculo del tanto por ciento de frecuencia de aparición.
4. Tener los datos disponibles en orden descendente de frecuencia.
5. Realizar el cálculo de las frecuencias acumuladas para cada causa.
6. Trazar dos ejes verticales y uno horizontal. El vertical derecho dibujarlo con una escala de 0-100% y el vertical izquierdo con el número de causas acumuladas. Después de esto, se reparte el eje horizontal en una cantidad de intervalos igual a la cantidad de elementos clasificados.
7. Realizar un gráfico de barras con idéntico ancho y sin dejar espacio entre ellas.
8. Dibujar la curva de frecuencias acumuladas
9. Escribir la información necesaria como título, unidades y periodo de tiempo entre otros.

12.2.1.4 Hoja de comprobación

¿Cuál es el principal objetivo del uso de las hojas de comprobación?

La hoja de comprobación se utiliza para reconocer y estudiar por una parte los problemas y por otra parte las causas. Con el fin de clasificar el conjunto de datos obtenidos según unas categorías específicas, se determina un conjunto de mecanismos necesarios para este fin. Todo ello será realizado mediante el apunte y registro de frecuencias para los contextos posibles.

Para conseguir este fin, por una parte es imprescindible detallar la estructura que se va a utilizar para guardar los datos, y por otro lado especificar el procedimiento que se va a utilizar para obtener y analizar los datos, señalando como, quien y cuando se va a realizar la planificación y la obtención.

¿Cómo se pueden clasificar las hojas de recopilación de datos?

Las hojas de recopilación de datos se pueden clasificar según el tipo de datos en los siguientes tipos:

- Clasificación: organización de artículos imperfectos o defectuosos.
- Verificación: engloba la comprobación, inspección, exploración o tarea de mantenimiento.

- Distribución: reparto de cambios de variables de los artículos. Algunos de ellos son por ejemplo el peso o la calidad entre otros.
- Localización: localización de anomalías en las piezas.

12.2.1.5 Diagrama de control

¿A que nos referimos cuando hablamos de diagramas de control?

Son representaciones gráficas usadas para analizar y por consiguiente determinar si un proceso está bajo control o no. ¿Qué quiere decir esto? Bien, pues un proceso estará bajo control siempre y cuando no exista una variabilidad, en caso de existir esta, se ha de analizar de donde proviene.

¿Qué se entiende por variabilidad?

Por variabilidad se entiende cualquier desviación existente en el producto con respecto a la especificación realizada inicialmente por los usuarios y la cual puede ser debida a alguno de los elementos.

¿Qué procedimiento de uso se sigue cuando se utiliza un diagrama de control?

Como se ha explicado en el primer punto, el principal objetivo de los diagramas de control consiste en saber si un proceso está bajo control o no. Es por esto, que el diagrama de control se utiliza para representar una determinada propiedad de calidad calculada a partir de muestras tomadas del producto a lo largo de un periodo de tiempo

¿Cómo es visualmente un diagrama de control?

Es un diagrama muy sencillo, el cual está formado por dos ejes de coordenadas, así como también por una línea central, la cual se suele posicionar en torno a la media muestral (μ), y dos límites de control, uno superior (LCS = Limite de control superior) y otro inferior (LCI=Limite límite de control inferior), que se fundamentan en conceptos y resultados estadísticos.

A continuación se incluye un ejemplo de diagrama de control, ya que de forma gráfica es más sencillo de comprender:

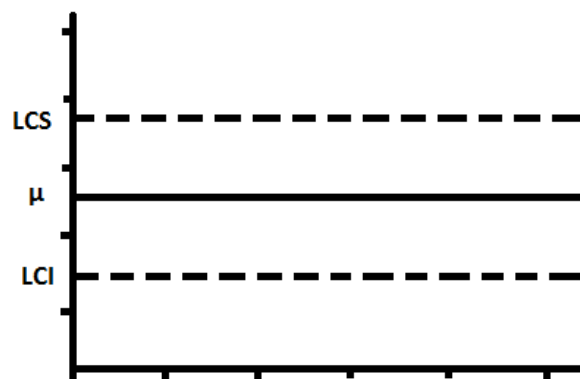


Ilustración 21: Diagrama de control tomado de [PGC06]

¿Qué tipos existen de diagramas de control?

Existen dos tipos de diagramas de control, los llamados “Gráficos de control por variables” y los “Gráficos de control por atributos”. No entramos en explicar en detalle cada uno de ellos, ya que no lo considero de gran importancia.

12.2.1.6 Histograma

¿A que nos referimos cuando hablamos de histogramas?

Igual que en el caso de los diagramas de control, es una representación gráfica mediante el uso de barras verticales que aporta información acerca de la frecuencia con la que suceden eventos enlazados entre sí.

Señalar que además de llamarse histograma, también en muchísimas ocasiones nos lo podemos encontrar con el nombre de “diagrama de distribución de frecuencias”.

¿Para qué es usado un histograma?

Básicamente, el histograma puede ser utilizado para los cuatro usos siguientes:

- Contrastar la variabilidad contra los límites de especificación.
- Conseguir una comunicación eficaz y clara de la variabilidad del sistema.
- Reconocer anomalías mediante el análisis de las diferentes formas del gráfico.
- Presentar el resultado de una modificación en el sistema.

¿Qué tipos de histogramas existen?

Existen diferentes tipos de histogramas dependiendo de la forma que estos tienen. Debido a que son números, a continuación se nombra alguno de ellos: histograma normal, histograma cortado, histograma bimodal e histograma distorsionado entre otros.

12.2.1.7 Diagrama de correlación

¿Para qué se utilizan los diagramas de correlación?

Como bien sabemos, es una herramienta que sirve para analizar una factible relación entre dos variables objeto de investigación de un control de calidad.

¿Cuál es el procedimiento que se sigue para utilizar el diagrama de correlación?

En un primer lugar, es necesario obtener un conjunto de datos de sucesos producidos donde colaboran los dos factores que se persigue analizar si existe una relación entre ellos. Los datos serán representados en el formato (x,y) donde la primera variable (x) está relacionada con el valor que se persigue determinar si contribuye (causa), mientras que la segunda (y), se corresponde con el valor de la consecuencia. El conjunto de datos se debe representar en un diagrama, y mediante el estudio de este se puede determinar si existe algún tipo de relación entre ambos factores.




12.2.2 Herramientas de gestión

12.2.2.1 Diagrama de afinidad

¿Para qué sirven los diagramas de afinidad?

En términos generales, los diagramas de afinidad valen para estructurar un conjunto de ideas en función de categorías vinculadas. Cabe señalar, que el conjunto de ideas mencionadas en la línea anterior, normalmente provienen de sesiones de tormentas de ideas.

¿Cuáles son los pasos a seguir para realizar un diagrama de afinidad?

-  Respecto al grupo de trabajo, anotar el conjunto de ideas y pensamientos que se manifiesten.
-  Crear categorías generales para el conjunto de ideas anteriores apoyándose en criterios de afinidad.
-  Realizar una correspondencia de cada idea a una de las categorías creadas en el punto anterior. Dicha asignación se debe realizar en función del grado de afinidad.

¿Qué tipos de diagramas de correlación existen?

Existen diferentes tipos de diagramas de correlación, los cuales se mencionan a continuación:

- ❖ No correlación
- ❖ Diagrama de “Correlación positiva”
- ❖ Diagrama de “Correlación negativa”
- ❖ Diagrama de “Correlación compleja”
- ❖ Diagrama de “Correlación fuertemente positiva”

Dependiendo de la forma resultante obtenida del conjunto de ideas, nos encontraremos en un tipo de diagrama de correlación u otro.

12.2.2.2 Diagrama de relaciones

¿En qué consisten los diagramas de relaciones?

Es una herramienta usada por una parte para reconocer las causas más importantes de un problema, y por otra parte para plasmar gráficamente las relaciones que puede haber entre los factores vinculados con ese problema.

¿Cuáles son los pasos a seguir para realizar un diagrama de relación?

Los pasos recomendados que hay que realizar para construir un diagrama de relación son cinco, y se nombran a continuación:

- Reconocer el conjunto de causas probables que dan lugar al problema.
- Formular o realizar una propuesta con una causa como la más factible.

- Analizar la relación existente entre la primera probable causa y el resto de causas, indicando mediante flechas el conjunto de relaciones que vayan apareciendo. Se puede dar el caso en el que existan distintos niveles de relación.
- Desechar en cada una de las iteraciones las causas que no han sido elegidas.
- Repetir la iteración con el fin de descubrir la causa que más conexiones o relaciones tenga.

12.2.2.3 Diagrama matricial

¿Para qué se utiliza el diagrama matricial?

En un diagrama matricial se representa gráficamente la relación que hay entre varios factores. Para este fin, basta con dibujar una matriz que contenga un conjunto de filas y un conjunto de columnas, donde en cada una de ellas se deben situar los factores. En caso de existir algún tipo de relación entre algunos de los factores, se debe marcar ésta en la intersección que forman ambos.

12.2.2.4 Matriz de análisis de datos

¿En qué consiste la matriz de análisis de datos?

En la mayoría de los casos, los autores definen esta herramienta como un subtipo del diagrama matricial. Esta herramienta también se puede denominar L-Shape, y en términos generales, lo que hace es enlazar dos conjuntos de elementos entre sí, aunque también en ocasiones llega a enlazar elementos del mismo conjunto.

12.2.2.5 Diagrama de redes de actividades

¿Qué son y para que se utilizan los diagramas de redes de actividades?

Igual que en las herramientas anteriores, esta consiste en una representación gráfica donde nos encontramos de una forma organizada el conjunto ordenado de tareas que hay que llevar a cabo en un plan de mejora de calidad.

¿Qué tipo de información aportan los diagramas de redes de actividades?

La información que debe aportar es la dependencia entre las distintas tareas o actividades, la holgura y por último la duración de cada tarea. Además el conjunto de todo el diagrama de red de actividades, también nos va a aportar información referente al tiempo necesario para llevar a cabo el plan de calidad. Probablemente nos formulemos la siguiente pregunta: ¿Cómo se calcula dicho tiempo? Bien, pues es posible obtener dicho valor ya que el diagrama de redes de actividades tiene un inicio y un final. Además, la unión entre las diferentes actividades en el diagrama se realiza mediante flechas, por lo que como estas muestran probables caminos, es factible reconocer caminos críticos en la ejecución del plan.

12.2.2.6 Diagrama de árbol

¿Para qué se usan normalmente los diagramas de árbol?

Primeramente indicar que un diagrama de árbol, para un producto concreto está formado por distintos niveles de complejidad, bien pues dicho esto, estos diagramas sirven para representar jerárquicamente dichos niveles iniciando desde un primer nivel y descomponiendo este en diferentes niveles con mayor grado de detalle hasta llegar a un nivel básico.

12.2.2.7 Diagrama de proceso de decisiones

¿En qué consiste el diagrama de proceso de decisiones?

Cuando nos encontramos con un problema, este tipo de diagramas define el plan de acciones a seguir con el objetivo de solucionar este. Normalmente, dichos planes suelen ser muy complejos.

¿Cuál es el procedimiento a seguir para crear un diagrama de proceso de decisiones?

Los pasos a seguir para la creación de un diagrama de este tipo son cinco, y se nombran a continuación:

1. Partiendo del plan planteado, elaborar un diagrama de árbol de tres niveles, donde cada nivel debe contener lo siguiente:
 - a. *Nivel 1*: propósito a conseguir con el plan.
 - b. *Nivel 2*: actividades importantes para alcanzarlo.
 - c. *Nivel 3*: listado de tareas a llevar a cabo para las actividades del nivel anterior.
2. Para las tareas definidas en el nivel 3, reconocer para cada una de ellas, que es lo que podría ocasionar problemas.
3. Verificar el conjunto de problemas de mayor importancia y suprimir aquellos en los que la probabilidad de que sucedan es nula o los efectos que pueden ocasionar tienen poca importancia. Por lo tanto en un cuarto nivel se pueden englobar estos problemas.
4. Para cada uno de los problemas de gran probabilidad o de mayor importancia, reconocer el conjunto de acciones que pueden aminorar los efectos causados por estos. Estos planes pueden ser incluidos en un quinto nivel.
5. Se analiza la viabilidad de cada uno de los planes anteriores, señalando con una "O" aquellos que pueden llegar a darse, y marcando con una "X" aquellos que son difíciles de darocurrir.

12.2.3 Herramientas de creatividad

¿Qué tipos de herramientas de creatividad existen?

Existe diversidad de herramientas de creatividad, pero en nuestro caso nos centraremos en la "tormenta de ideas", ya que esta es la más utilizada.

¿En qué consiste la "tormenta de ideas"?

El principal objetivo de esta técnica consiste en conseguir en el menor tiempo posible el mayor número de soluciones o como su nombre indica de ideas, eligiendo a posteriori aquellas que mejor se ajustan al propósito del problema. Por lo tanto, del conjunto de personas que forman el grupo de trabajo, lo que más se valora es la creatividad de cada uno de ellos.

¿Cuáles son los modos existentes para llevar a cabo dicha técnica?

Existen dos modos conocidos para llevar a cabo la tormenta de ideas: modo estructurado y modo libre. La diferencia entre ambos radica que en el primer modo existe un orden riguroso de participación, mientras que en el segundo, cada persona aporta sus ideas según se les van ocurriendo, sin seguir orden alguno. El segundo modo, en mi opinión es mejor opción, ya que al no estar presionado bajo ningún control, la aportación es mejor y se realiza de manera más relajada.

12.2.4 Herramientas estadísticas

12.2.4.1 Control estadístico del proceso

¿En qué se basa esta herramienta de control estadístico del proceso?

Para entender esto, lo primero que debemos saber es el significado de capacidad de un proceso. Pues bien este término está relacionado con el grado de capacidad que posee para cumplir con el conjunto de especificaciones técnicas que se solicitan. Llegados hasta aquí, cuando esta capacidad de la que estamos hablando es alta, entonces es capaz de satisfacer los requisitos técnicos si por el contrario esta se mantiene constante en el tiempo se dice que está bajo control.

¿Qué herramientas pueden ser utilizadas para el cálculo de la capacidad?

Para concretar si un proceso es capaz o por el contrario no lo es de satisfacer las especificaciones técnicas, se pueden usar un conjunto de herramientas como son histogramas, estudios de índices de capacidad, gráficos de control y gráficos de probabilidad.

12.2.4.2 Diseño de experimento

¿Cuál es el principal objetivo de esta herramienta?

Esta herramienta se centra en indagar si algunos factores afectan sobre determinadas variables de gran interés para calidad, y en caso de influir, cuantificarla.

¿Esta herramienta puede ser llamada de alguna otra forma?

Sí en este caso, aparte de llamarse diseño de experimentos, puede ser nombrado como DOE (Design of Experiments)

¿De qué etapas está formado un DDE?

Las etapas de las que está compuesto un DDE son nueve y todas ellas se mencionan a continuación:

1. Definir los objetivos o propósitos del experimento.
2. Reconocer los posibles motivos de variación.
3. Escoger el diseño experimental apropiado.
4. Detallar las medidas así como también el procedimiento experimental.
5. Realizar un experimento piloto.
6. Detallar el tipo de modelo.
7. Representar mediante esquemas cada uno de los pasos del análisis estadístico.

8. Establecer el tamaño de la muestra.
9. Chequear el conjunto de decisiones tomadas en los pasos anteriores.

12.2.5 Herramientas de diseño

12.2.5.1 Quality function deployment (QFD)

¿Para qué se utiliza esta técnica?

Esta herramienta, mediante la utilización de métodos matriciales se usa para proyectar nuevos productos o así como también para perfeccionar productos ya existentes.

¿Cuál es el principal objetivo que se persigue?

El principal objetivo que se persigue, es que el conjunto de requisitos definidos por el cliente estén totalmente incluidos en las especificaciones técnicas del producto.

¿Qué ventajas se tiene si se usa esta herramienta?

Por una parte se minimiza el tiempo empleado en el diseño, y por otra se realiza una reducción de los costes, siempre conservando y progresando la calidad.

¿Qué pasos se deben seguir para realizar un proyecto usando esta herramienta?

Para la realización de un proyecto usando “Quality Function Deployment”, se debe pasar por tres fases claramente diferenciadas:

- ❖ **Fase 1:** la primera fase es la llamada “fase de organización”, y donde se realizará una acotación del alcance del proyecto, especificando el propósito del proyecto así como también el conjunto de individuos que deben participar en el proyecto. Este conjunto de personas deben cumplir ciertos requisitos, como poseer experiencia y estar asociados a diferentes departamentos conectados con el proyecto.
- ❖ **Fase 2:** es la conocida como “fase de definición”, y en la que se realiza básicamente una planificación provisional de la duración de cada tarea y del orden de estas. Además, es de gran importancia revisar el propósito del proyecto para adecuarlo a los recursos disponibles de la compañía.
- ❖ **Fase 3:** es la nombrada como “Fase de identificación y análisis de necesidades” y donde lo que se hace es tomar los requisitos aportados por el cliente, estudiarlos y los distintos individuos que forman el grupo de trabajo deben entenderlos, para que finalmente se realice una relación entre las características del producto y el conjunto de requisitos, ya que ambos deben sintonizar. Para la realización de esto anterior, se utilizan normalmente cuatro tipos de matrices, y son las llamadas “Matriz de planificación del producto”, “Matriz de planificación del proceso”, “Matriz de despliegue de componente” y “Matriz de la planificación de la producción”. No entraremos en detalle en cada una de ellas, ya que no lo considero de especial interés dentro de este proyecto.

12.2.5.2 Análisis modal de fallos y efectos (AMFE)

¿En qué consiste el “Análisis modal de fallos y efectos”?

Es un proceso sistemático, elaborado y participativo que se lleva a cabo o bien cuando hay nuevos productos, o bien cuando se hacen cambios de gran peso con el fin de valorar o descubrir fallos que son producidos con anterioridad a que el producto llegue al cliente.

¿Qué se hace con el conjunto de fallos obtenidos?

Del conjunto de fallos obtenidos, se da prioridad a cada uno de ellos dependiendo de lo siguiente:

- Frecuencia de aparición.
- Gravedad de las consecuencias que generan.
- Facilidad de detección de los fallos.

¿Qué proporciona esta herramienta?

Este método nos da la opción de aminorar tiempos, también mejorar y determinar un contexto de aseguramiento continuo de la calidad y por último incrementar la fiabilidad de los productos.

¿Cuál es el procedimiento que se sigue para el desarrollo de un AMFE?

Para el desarrollo del análisis modal de fallos y efectos, basta con seguir los siguientes cinco pasos:

1. Formar un grupo de personas multidisciplinar con un entendimiento amplio y variado sobre diferentes factores referentes a los usuarios como son productos, procesos, servicios y necesidades de los usuarios. Además de esto anterior, estos deberían poder llevar a cabo otras actividades como son diseño, calidad y pruebas entre otros.
2. Realizar una valoración por una parte del alcance, y por otra de los límites de aplicación del análisis modal de análisis y fallos, reconociendo el proceso a analizar, así como también los modos de fallos, causas y finalmente los efectos.
3. Hacer y completar el conjunto de documentación referente a la valoración de precedencia para cada uno de los modos de fallo. Como rellenar la documentación implica también rellenar una tabla, para ello es necesario realizar los cálculos correspondientes de determinados índices entre ellos el IPR.
4. Dependiendo del valor del IPR, valorar las acciones correctivas a llevar a cabo, seleccionar a los responsables y realizar una estimación del plazo necesario para llevarlo a cabo.
5. Llevar a cabo las acciones correctivas anteriores y una vez sobrepasada la fecha de aplicación, recalcular los índices con el fin de verificar la vigencia de las acciones correctivas efectuadas.

12.2.6 Herramientas de medición

12.2.6.1 Coste de la calidad (COQ)

¿En qué consiste la herramienta de coste de la calidad?

Análisis de requisitos en el desarrollo del software

Es un proceso usado para reconocer posibles problemas y realizar un cálculo de los costes que originaría el no realizar las cosas correctamente desde un principio. En muchas ocasiones esta herramienta también es denominada como “Análisis de costes de pobre calidad”.

¿Qué procedimiento se ha de seguir para realizar un análisis de coste de la calidad?

Para la realización de un análisis de coste de la calidad, es conveniente seguir los siguientes cinco pasos:

1. Adquirir o realizar un dibujo de un diagrama de flujo minucioso del proceso.
2. Reconocer el conjunto de fases y tareas que forman el proceso y señalar aquellas que incurran en los costes de calidad: observación, arreglo y seguimiento de daños. Debater acerca de las razones de que existan muchas o pocas tareas señaladas que incurran en los costes de calidad.
3. Para cada una de las tareas señaladas en la fase anterior, calcular el coste que pueden suponer los fallos provenientes de una baja calidad y también el coste que puede conllevar el aplicar el conjunto de acciones correctivas.
4. Calcular como son de viables las acciones tomadas en el punto anterior.
5. Sugerir aquellas acciones correctivas en las que la viabilidad sea posible.

12.2.6.2 Benchmarking

¿Cómo se define la herramienta “Benchmarking”?

Según “Tague”, es un proceso estructurado que da la posibilidad de realizar una comparación de las mejores prácticas utilizadas por las diferentes sociedades, con el objetivo de poder añadir aquellas prácticas que perfeccionan las que se desarrollan para la sociedad, o a los procesos de esta o finalmente las que no se desarrollan.

¿Cuáles son las fases a seguir para el desarrollo de un “Benchmarking”?

Debido a que para el desarrollo de un “Benchmarking” las fases a realizar son cuatro y cada una de ellas engloba una serie de acciones a seguir, es por esto, que a continuación dicha información será mostrada en un cuadro resumen con el objetivo de poder visualizar el conjunto de información de manera más cómoda.

FASE	ACCIONES QUE ENGLOBA CADA FASE	
Planificar	Acción 1	Del conjunto de objetivos críticos que intervienen en el éxito de la organización, especificar cada uno de ellos.
	Acción 2	Juntar a un grupo de personas para formar un equipo que haga frente al estudio de los objetivos que se van a llevar a cabo.
	Acción 3	Analizar minuciosamente cada uno de los procesos de la organización, ya que, es necesario saber cómo funcionan en una organización las cosas internamente para poder realizar un correcto trabajo en la comparación.

	Acción 4	Del conjunto de técnicos que forman la organización, seleccionar aquellos que podrían llevar a cabo el desarrollo de las mejores prácticas.
Recopilar Datos		Como su nombre indica, esta fase consiste en reunir el conjunto de información proveniente de los técnicos de la organización. La información será recopilada mediante el uso de entrevistas, cuestionarios y será información relativa a la especificación de los procesos así como a datos numéricos.
Analizar	Acción 1	De los datos anteriormente capturados, realizar una comparación de estos.
	Acción 2	Establecer las brechas entre las medias de rendimiento de otras sociedades con las de la propia sociedad.
	Acción 3	Establecer las desigualdades en las prácticas que provocan las brechas anteriores.
Adaptar	Acción 1	Establecer objetivos para los diferentes procesos de la organización.
	Acción 2	Realizar el desarrollo de los planes de acción con el fin de lograr los objetivos.
	Acción 3	Efectuar los planes de acciones anteriores y monitorizarlos.

Tabla 22: Fases de desarrollo de un "Benchmarking"

12.2.6.3 Encuestas

¿Cuál es la principal finalidad de las encuestas?

El principal objetivo de las encuestas es establecer la naturaleza de los procesos. Debido a que estas están explicadas en detalle en apartados anteriores, aquí no nos detendremos en ellas.

12.3 Normas y estándares de calidad

A lo largo de todo el proyecto se han hecho diferentes menciones a normas ISO relacionadas con la ingeniería del software.

Las siglas ISO se corresponden con la expresión "International Organization for Standardization" (Organización Internacional de Estandarización). Como su nombre indica el principal objetivo de dicha organización es simplificar la coordinación internacional de las normas.

Hay que tener en cuenta que el tiempo que transcurre desde que una norma se elabora hasta que se realiza la publicación final en situaciones normales es muy largo. En España, la organización que se encarga de la traducción y publicación de las normas es AENOR.

12.3.1 Normas relacionadas con calidad

En este apartado nos encontraremos infinidad de ocasiones la mención a las siglas UNE, la traducción de dichas siglas se corresponde con "Una norma española".

Análisis de requisitos en el desarrollo del software

En este apartado hablaremos especialmente de las normas ISO 9000, debido a que estas son las que mayor grado de difusión han tenido. La publicación inicial de estas fue hecha en el año 1987, y como protocolo a seguir en todas las normas ISO, esta ha sido revisada al menos una vez cada cinco años. La ISO 9000 la forman cuatro normas que se comentan a continuación:

UNE-EN ISO 9000	<i>Sistemas de gestión de la calidad. Fundamentos y vocabulario</i>	Como se puede intuir por el nombre de la norma, esta especifica una parte de los fundamentos de los sistemas de gestión de la calidad y por otra parte describe su terminología.
UNE-EN ISO 9001	<i>Sistemas de gestión de la calidad. Requisitos.</i>	En esta norma nos podemos encontrar una especificación de los requisitos a poder ser utilizados por las organizaciones para un sistema de gestión de calidad, bien para certificación o bien para otros fines. Principalmente analiza como de eficaz es el sistema de gestión de calidad para proporcionar cumplimiento al grupo de requisitos del cliente.
UNE-EN ISO 9004	<i>Sistemas de gestión de la calidad. Directrices para la mejora del desempeño.</i>	Esta norma aporta una orientación más amplia de propósitos de un sistema de gestión de calidad, específicamente con el fin de mejorar la eficiencia de la organización. En la mayoría de las ocasiones esta norma se aconseja utilizar en los casos en los que el principal objetivo de las organizaciones es la mejora continua del desempeño.
UNE-EN ISO 19011	<i>Directrices para la auditoría de sistemas de gestión de la calidad y/o medioambiental.</i>	Esta norma aporta pautas básicas para la ejecución de una auditoría conjunta de ISO 14001 (sistemas de gestión medioambiental) e ISO 9001.

Tabla 23: Conjunto de normas que forman la ISO 9000

Además, existe otro conjunto de normas también relacionadas con calidad y con las normas ISO 9000, y las cuales se muestran en la tabla siguiente:

NORMA ISO	NOMBRE DESCRIPTIVO
ISO 10002	Quality management- Customer Satisfaction (Satisfacción del cliente. Gestión de reclamaciones)
ISO 10005	Quality Plan (Planes de calidad)
ISO 10006	Directrices para la calidad en la gestión de proyectos
ISO 10007	Directrices para la gestión de la configuración
ISO 10012	Sistemas de gestión de las mediciones
ISO/TR 10013	Directrices para la documentación de sistemas de gestión de la calidad
ISO/TR 10014	Directrices para la gestión de los efectos económicos de la calidad
ISO/IEC 2500n	División de Gestión de Calidad
ISO/IEC 2501n	División de Modelo de Calidad
ISO/IEC 2502n	División de Medición de Calidad
ISO/IEC 2503n	División de Requisitos de Calidad
ISO/IEC 2504n	División de Evaluación de Calidad

Tabla 24: Normas relacionadas con calidad y con las normas ISO 9000 adaptado de [PGC06]

12.3.2 Principios relacionados con las normas ISO9000

Existe un conjunto de principios básicos relacionados con la gestión de calidad que la dirección puede usar con el principal objetivo de guiar a la organización hacia una mejora en la ejecución del desempeño.

El número de principios básicos de los que hablaremos suman un total de nueve, y estos se explican a continuación:

Principio 1: Mejora continua

La mejora continua de la función a desempeñar por la organización siempre debe ser un propósito permanente a alcanzar por esta.

Principio 2: Participación del personal

El personal es uno de los puntos fuertes dentro de una organización, hablando desde los niveles más bajos a los niveles más altos, es por ello, que su total compromiso con la organización facilita que la destreza y habilidad de cada uno de ellos sean utilizadas para el beneficio de esta.

Principio 3: Enfoque al cliente

Todas las organizaciones presentan una dependencia de sus clientes, ya que si estos les fallan, la organización fracasa. Bien, pues teniendo esto en mente, siempre la organización debería entender las necesidades de estos, tanto actuales como posibles necesidades futuras, satisfacer sus requisitos así como también intentar superar sus expectativas.

Principio 4: Enfoque basado en hechos para la toma de decisión

El conjunto de decisiones efectivas tomadas se fundamentan principalmente por una parte en el análisis de datos y por otra en la información.

Principio 5: Liderazgo

Los líderes son los encargados de fijar los objetivos a conseguir y fijar la dirección de la organización. Es por esto, que ellos son los encargados de crear y sostener un ambiente adecuado con el fin de que los empleados de la organización lleguen a implicarse completamente para conseguir los propósitos perseguidos.

Principio 6: Relaciones mutuamente beneficiosas con el proveedor

Siempre existe una relación de dependencia entre la organización y los proveedores, por lo que cuanto mejor sea la relación entre ambos mayor será la capacidad para crear valor.

Principio 7: Enfoque de sistema para la gestión

Reconocer, comprender y tramitar los diferentes procesos interconectados, ayuda a que factores como la eficacia dentro de una organización haga alcanzar los propósitos fijados.

Principio 8: Enfoque basado en procesos

Este conjunto de normas fomenta el apadrinamiento de un enfoque basado en procesos en las fases de desarrollo, implementación y mejora de un sistema de gestión de calidad.

12.4 Calidad de productos software

Existen un conjunto de modelos, cuya funcionalidad es realizar una valoración de la calidad de los productos software. A continuación se muestran algunos de ellos:

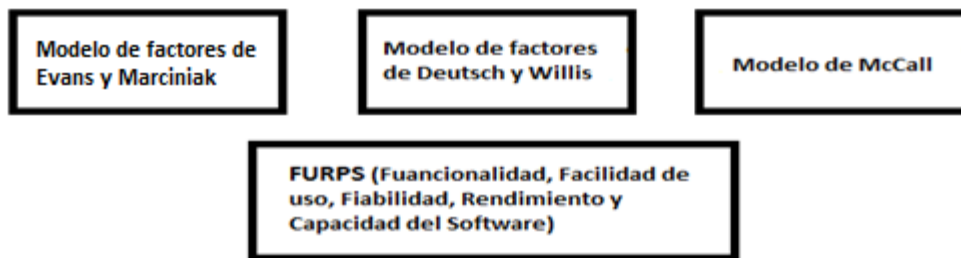


Ilustración 22: Modelos utilizados para valorar la calidad de productos software

De todos los existentes, uno de los más usados hasta la actualidad, fue creado por McCall, el cual se fundamenta en que existen once factores de calidad clasificados en tres grupos que forman la calidad de un producto software. Estos tres amplios grupos a los que hacemos referencia son: operación de producto, revisión de producto y por último transición de producto.

Aunque se ha mencionado anteriormente que el modelo de McCall es el más utilizado, en el pasado se realizó una comparación de tres modelos (McCall, Evans y Marciniak, Deutsch y Willis) en base a dieciséis factores de calidad (corrección, fiabilidad, eficiencia, integridad, usabilidad, mantenibilidad, flexibilidad, testeabilidad, portabilidad, reusabilidad, interoperabilidad, verificabilidad, expandibilidad, seguridad de uso, manejabilidad y capacidad de supervivencia), y donde se pudo observar que el modelo más completo era el de Deutsch y Willis ya que este satisfacía todos los factores de calidad excepto el de testeabilidad.

12.4.1 Aspectos a destacar de la calidad de un producto software

A lo largo de las diferentes etapas del ciclo de vida del software, en la calidad de un producto software, normalmente se pueden diferenciar tres aspectos, los cuales son los siguientes:

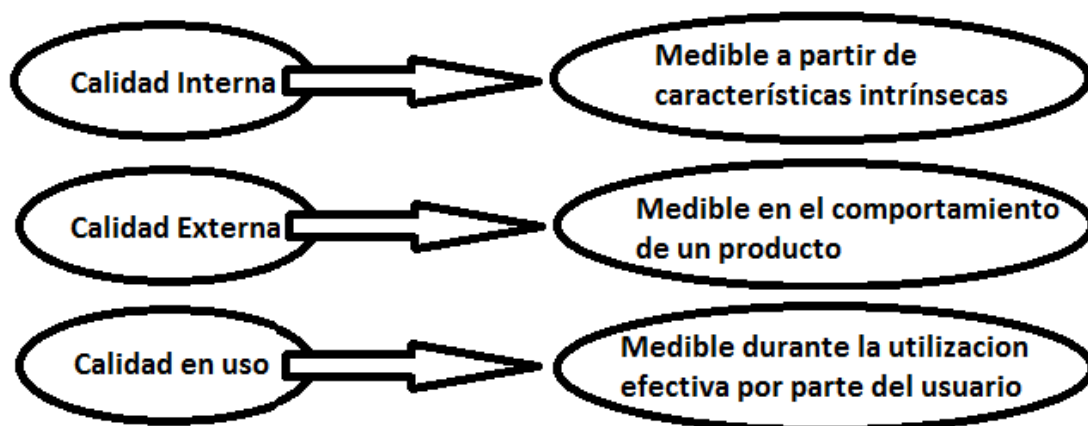


Ilustración 23: Tipos de calidad

12.4.1.1 Modelo de calidad externa e interna

Según ISO 2001 este modelo categoriza los factores de calidad software en seis distintas características, las cuales a su vez se desglosan en subcaracterísticas.

Análisis de requisitos en el desarrollo del software

Las seis características de las que hablamos son: funcionalidad, usabilidad, mantenibilidad, fiabilidad, eficiencia y portabilidad. Posteriormente se va a realizar una descripción escueta de cada una de ellas con el fin de adquirir una visión global de ellas.

CARACTERISTICA	DESCRIPCION	SUBGRUPOS / DESCRIPCION	
FUNCIONALIDAD	Habilidad del producto software para suministrar un conjunto de funciones que utilizadas bajo unas determinadas condiciones, satisface las necesidades requeridas.	Adecuación	Habilidad del producto software para suministrar un conjunto adecuado de funciones para tareas y propósitos de usuario determinados.
		Exactitud	Habilidad del producto software para suministrar los resultados con el grado de precisión necesario.
		Interoperabilidad	Habilidad del producto software para interactuar con determinados sistemas.
		Seguridad de acceso	Habilidad del producto software para salvaguardar datos e información de tal manera que los individuos no autorizados para ello no lleguen a leerlos ni modificarlos, de la misma manera, que a personas autorizadas para ello se les permita.
		Cumplimiento funcional	Habilidad del producto software para unirse a normas y regulaciones entre otras vinculadas con funcionalidad.

Tabla 25: La funcionalidad como factor de calidad externa e interna

CARACTERISTICA	DESCRIPCION	SUBGRUPOS / DESCRIPCION	
FIABILIDAD	Habilidad del producto software, que utilizado bajo unas determinadas condiciones, sostiene un nivel explícito de prestaciones	Madurez	Habilidad del producto software para prescindir de fallos provenientes del software.
		Tolerancia a fallos	En el supuesto caso de fallos en el software, esta característica es una habilidad del producto software para conservar un determinado nivel de prestaciones.
		Capacidad de recuperación	Esta referidaa la capacidad del producto software para en caso de fallo volver a la situación normal.
		Cumplimiento de la fiabilidad	Habilidad del producto software para unirse a normas y regulaciones entre

			otras vinculadas con la fiabilidad.
--	--	--	-------------------------------------

Tabla 26: La fiabilidad como factor de calidad externa e interna

CARACTERISTICA	DESCRIPCION	SUBGRUPOS / DESCRIPCION	
USABILIDAD	Utilizado bajo unas determinadas condiciones, se refiere a la habilidad del producto software para ser comprendido, atractivo y utilizado para el usuario.	Capacidad para ser entendido	Bajo unas condiciones determinadas, la capacidad para ser entendido es una habilidad que concede al usuario comprender si el software es adecuado, y la manera en la que puede ser utilizado este para las diferentes funciones.
		Capacidad para ser aprendido	Concede al usuario del producto software la posibilidad de coger recibir una formación acerca de su aplicación.
		Capacidad para ser operado	La capacidad para ser operado, concede al usuario del producto software dominar y operar este.
		Capacidad de atracción	Habilidad del producto software con el fin de ser atractivo de cara al usuario de este.
		Cumplimiento de la usabilidad	Habilidad del producto software para unirse a normas y regulaciones entre otras vinculadas con la usabilidad.

Tabla 27: La usabilidad como factor de calidad externa e interna

CARACTERISTICA	DESCRIPCION	SUBGRUPOS / DESCRIPCION	
EFICIENCIA	Utilizado bajo unas condiciones determinadas, dicha característica es una habilidad del producto software que suministra prestaciones adecuadas relacionadas al número de recursos utilizados.	Comportamiento temporal	Bajo condiciones específicas, es una habilidad relacionada con el producto software que consiste en aportar tiempos de respuesta, proceso y potencia adecuados.
		Utilización de recursos	Cuando se utilizan condiciones específicas, consiste en la habilidad del producto software para utilizar el número y tipo de recursos apropiados.
		Cumplimiento de la eficiencia	Habilidad del producto software para unirse a normas y regulaciones entre otras vinculadas con la eficiencia.

Tabla 28: La eficiencia como factor de calidad externa e interna

CARACTERISTICA	DESCRIPCION	SUBGRUPOS / DESCRIPCION	
MANTENIBILIDAD	En términos generales es una habilidad para que el producto software sea rectificado. Estas rectificaciones engloban modificaciones y cambios entre otros así como requisitos y especificaciones funcionales.	Capacidad para ser analizado	Es la habilidad del producto software para que deficiencias o fallos sean detectados, así como también para reconocer las distintas partes que tienen que ser rectificadas.
		Capacidad para ser cambiado	En este caso, la capacidad para ser cambiado permite que las modificaciones específicas en productos software sean implementadas.
		Estabilidad	Debido a que las modificaciones de software pueden generar efectos inesperados, la estabilidad en un producto software evita dichos efectos.
		Capacidad para ser aprobado	Una vez que el software ha sufrido modificaciones, la capacidad para ser probado es una habilidad del producto software que nos da la opción de que este sea verificado así como también validado.
		Cumplimiento de la mantenibilidad	Habilidad del producto software para unirse a normas y regulaciones entre otras vinculadas con la mantenibilidad.

Tabla 29: La mantenibilidad como factor de calidad externa e interna

CARACTERISTICA	DESCRIPCION	SUBGRUPOS / DESCRIPCION	
PORTABILIDAD	Como su nombre indica consiste en la capacidad para que un producto sea portado o transferido entre diferentes entornos.	Adaptabilidad	Capacidad para que un producto software sea adaptado a determinados entornos.
		Instalabilidad	Habilidad del producto software que permite que este sea instalado en un entorno determinado.
		Coexistencia	Habilidad del producto software para convivir con otro software, con recursos compartidos y en un entorno común.
		Capacidad para ser reemplazado	Persiguiendo el mismo objetivo y con idéntico entorno, la capacidad para ser reemplazado es una habilidad del producto software para ser utilizado

			en el lugar de otro.
		Cumplimiento de la portabilidad	Habilidad del producto software para unirse a normas y convenciones entre otras vinculadas con la portabilidad.

Tabla 30: La portabilidad como factor de calidad externa e interna

12.4.1.2 Modelo de calidad en uso

La definición de “Calidad en uso” según la norma ISO 9126 es la siguiente:

“La capacidad del producto software para permitir a determinados usuarios alcanzar objetivos especificados con efectividad, productividad, seguridad y satisfacción, en contextos de usos especificados”.

Igual que ocurre con el modelo de calidad externa e interna, la calidad en uso se subdivide en cuatro características (efectividad, productividad, seguridad de uso y satisfacción), las cuales serán explicadas de forma escueta a continuación.

CARACTERISTICA	DESCRIPCION
EFFECTIVIDAD	En un contexto de uso determinado, dicha característica consiste en una capacidad relacionada con el producto software para permitir conseguir determinados propósitos con exactitud al conjunto de usuarios.

Tabla 31: La efectividad como factor de calidad en uso

CARACTERISTICA	DESCRIPCION
PRODUCTIVIDAD	En un contexto de uso determinado, la productividad se refiere a una habilidad del producto software para permitir desembolsar ¿dedicar / aplicar? una apropiada cantidad de recursos con relación a la eficacia alcanzada a los usuarios.

Tabla 32: La productividad como factor de calidad en uso

CARACTERISTICA	DESCRIPCION
SEGURIDAD DE USO	En un contexto de uso determinado, capacidad relacionada con el producto software para conseguir niveles admisibles de causar riesgos entre otros al software o a personas.

Tabla 33: La seguridad en uso como factor de calidad en uso

CARACTERISTICA	DESCRIPCION
SATISFACCION	Como su nombre indica, en un contexto de uso determinado, es una capacidad relacionada con el producto software para satisfacer a los distintos usuarios.

Tabla 34: La satisfacción como factor de calidad en uso

12.5 Caso práctico: Encuesta de calidad

12.5.1 Introducción caso práctico

Como adjunto a este proyecto, se ha realizado una encuesta sencilla, programada en el lenguaje de programación HTML, que engloba un conjunto de cuestiones cuyo fin principal es el de evaluar el sistema con el fin de verificar la calidad de este.

Me he decantado por utilizar de ejemplo esta técnica ya que actualmente es la más utilizada. Cabe destacar que el modelo EFQM considera esto como el criterio de mayor peso, y por otra parte la norma ISO 9001 lo incluye como requisito. Otro aspecto a tener en cuenta, es que la satisfacción del cliente es el principal indicador para realizar una evaluación del trabajo de la organización.

La primera pregunta que me hice antes de hacer esto fue: ¿Qué me aporta la realización por parte de los agentes de la encuesta de calidad? Bien, pues estudiando esto llegué a la conclusión de que esta me aportaría información de carácter importante que me permitiría realizar acciones que mejorarían mi sistema.

La encuesta, como bien sabemos está formada por un conjunto de secciones que a su vez cada una de estas lo forman un conjunto de preguntas. Lo complicado es redactar el conjunto de preguntas, ya que estas deben ser redactadas de forma estratégica, para que podamos obtener la información deseada, aparte de que es de gran importancia identificar el conjunto de características que el cliente valora.


En un primer momento analicé que tipo de información debía de obtener para verificar ciertos aspectos de mi sistema relacionados con calidad, algunos de estos son los siguientes:

- En primer lugar saber si mi sistema es de buena calidad.
- Saber si mi sistema posee todas las características y funciones que el cliente necesita.
- Saber si mi sistema es fácil de utilizar.
- Saber si mi sistema está disponible las 24 horas del día y los 365 días al año.
- Saber de algún modo si mi sistema tiene defectos o imperfecciones.
- Conocer si los agentes desean que mi sistema cambie o tenga alguna funcionalidad más.

12.5.2 Contenido de la encuesta de calidad

Para obtener todo este tipo de información, como se ha mencionado anteriormente, se han formulado un conjunto de preguntas. A continuación se adjuntan capturas correspondientes a la encuesta de calidad realizada:

Este primer grupo de cuatro preguntas se realiza con el fin de conocer el tipo de agente que es, es decir, no es lo mismo la opinión aportada por un agente con gran cantidad de años de experiencia y gran cantidad de unidades ofertantes, que la de un agente con poca experiencia y con un número pequeño de unidades.



ENCUESTA DE SATISFACION DE LOS AGENTES


Esta encuesta se realiza con motivo de conocer la opinión general sobre la operativa del web y el servicio que prestamos al conjunto de agentes. Dicha encuesta es anónima, por lo que los comentarios vertidos en esta no tendrán ningún tipo de repercusión.

1. ¿Desde qué año presenta la condición de agente?
 - ☒ Desde el año 2006 o antes
 - ☐ Entre el año 2006 y 2014
 - ☐ Después del año 2014
2. Media de volumen anual que resulta casado en nuestra compañía.
 - ☐ Menos de 5GWh
 - ☒ Más de 5GWh
3. ¿Cuántas UOF tiene dadas de alta para ofertar?
 - ☐ Menos de 5 unidades
 - ☐ Más de 5 unidades
4. Número medio de ofertas que presenta en el mercado Intradiario.
 - ☒ Menos de 5 ofertas
 - ☐ Más de 5 ofertas

[siguiente](#)

Dalila García

La finalidad de este siguiente grupo formado por otras cuatro preguntas, es la de conocer si las principales funciones del sistema funcionan de una manera correcta. Cabe mencionar que el comienzo y fin de una sesión, cualquier tipo de notificación, como por ejemplo el retraso en un mercado, junto con la publicación de resultados, son los aspectos más delicados, los que más se cuidan y los que más se analizan en la compañía.



ENCUESTA DE SATISFACION DE LOS AGENTES

INDICADORES OBJETIVOS

Las siguientes preguntas están relacionadas con diferentes aspectos de la operativa de ambos mercados.
Para cada pregunta se debe marcar la opción que más se ajuste a su situación.

5. ¿Todas las sesiones comienzan en su debido tiempo?

☐ Siempre

☒ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

6. ¿Todas las sesiones finalizan en su debido tiempo?

☒ Siempre

☐ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

7. En caso de tener que notificar alguna noticia, ¿Dicha información es recibida a tiempo y en la forma correcta?

☐ Siempre

☐ Casi Siempre

☒ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

8. ¿Los resultados obtenidos de la casación se comunican de manera correcta?

☒ Siempre

☐ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

[Anterior](#) [Siguiente](#)

El siguiente grupo de dos preguntas está relacionado con el departamento de sistemas de información (DSI), y por consiguiente se comprueba si el acceso al sistema y la velocidad de respuesta en caso de problema es la oportuna, así como también se pretende verificar la sencillez de este.

ENCUESTA DE SATISFACION DE LOS AGENTES

SOBRE EL SISTEMA DE INFORMACION Y SOFTWARE

9. ¿El acceso a los sistemas de la compañía es sencillo?

☐ Siempre

☐ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

10. En caso de problema o necesidad de ayuda, ¿La velocidad de respuesta a la solicitud es la correcta?

☐ Siempre

☒ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente


☐ No aplica

Anterior Siguiente

Dalila Garcia

El siguiente grupo de tres preguntas está relacionado con el departamento de liquidaciones, y por consiguiente se pretende saber si la descarga de las facturas así como la información de gran importancia se puede realizar de forma sencilla y en los plazos establecidos por la compañía.

Así a primera vista, parece que esto carece de importancia, pero no es así, ya que una errónea información de liquidaciones puede llevar a que la compañía rechace una oferta de un agente de envergadura, por lo que es de gran importancia que todos los apuntes de garantías sean los correctos y estén disponibles en la fecha estimada.



The screenshot shows a web browser window with the address bar displaying 'IR_AQU\CUESTIONARIO\DEFINITIVO_ENTREGAR' and a tab titled 'ENCUESTA DE SATISFACION...'. The page features a light blue header with a globe icon. The main content area is white and contains the title 'ENCUESTA DE SATISFACION DE LOS AGENTES' in blue. Below this is a section titled 'SOBRE EL SISTEMA DE LIQUIDACIONES Y FACTURACION' in orange. The survey consists of three questions, each with five radio button options: 'Siempre', 'Casi Siempre', 'Normalmente', 'A veces', and 'No aplica'. Question 11 asks about the timeliness of liquidations and invoicing. Question 12 asks about the simplicity of access to liquidation systems. Question 13 asks about the simplicity of downloading information related to invoices and liquidations. The third question and its options are highlighted with a yellow background. At the bottom, there are two blue buttons labeled 'Anterior' and 'Siguiente', and the name 'Dalila García' is displayed.

ENCUESTA DE SATISFACION DE LOS AGENTES

SOBRE EL SISTEMA DE LIQUIDACIONES Y FACTURACION

11. ¿Obtiene las liquidaciones y facturaciones en el plazo establecido?

☒ Siempre

☐ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

12. ¿El acceso a los sistemas de liquidaciones es sencillo?

☐ Siempre

☒ Casi Siempre

☐ Normalmente

☐ A veces

☐ Normalmente

☐ No aplica

13. ¿Se descarga de forma sencilla la información referente a facturas y liquidaciones?

☒ Siempre

☐ Casi Siempre

☐ Normalmente

☐ A veces

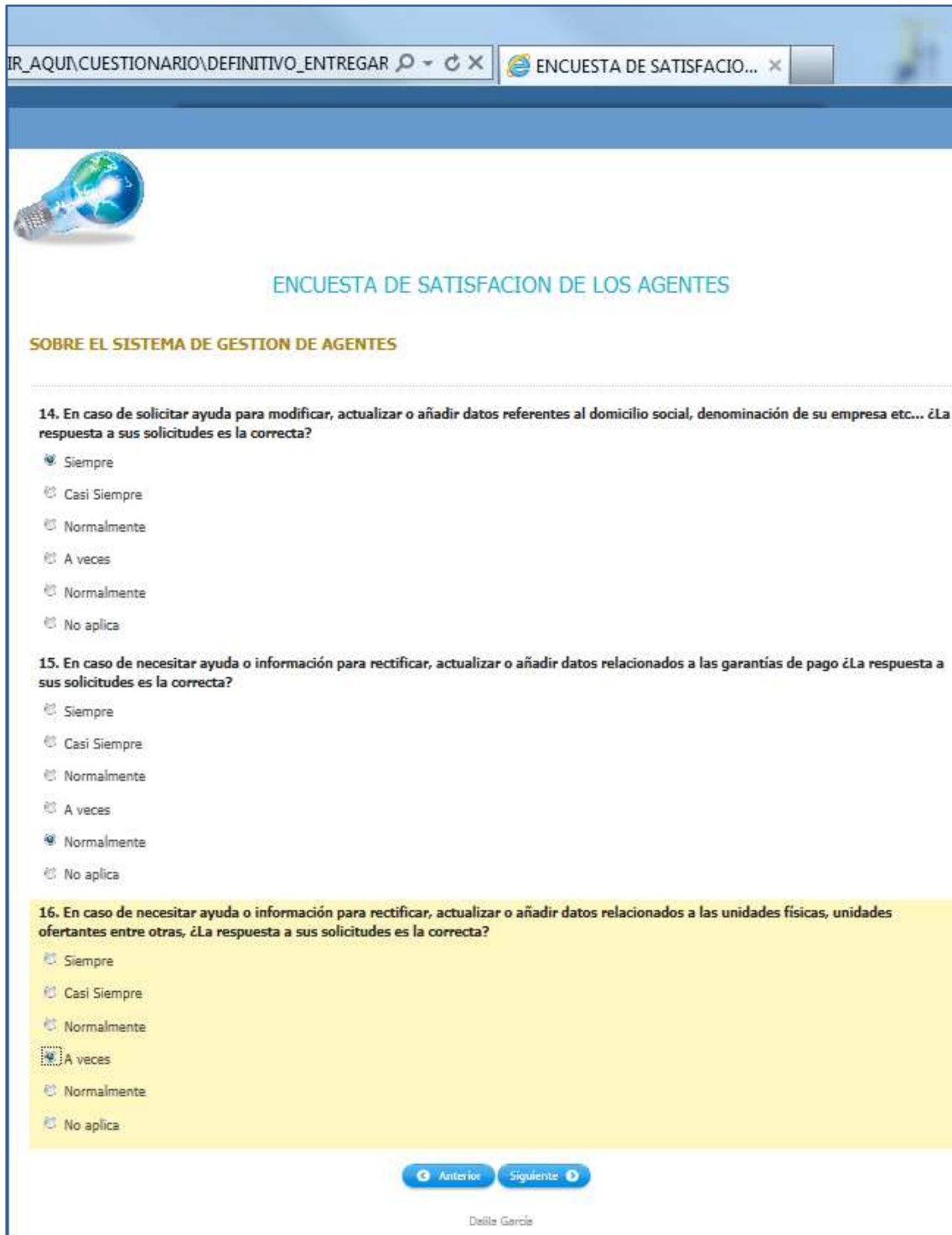
☐ Normalmente

☐ No aplica

[Anterior](#) [Siguiente](#)

Dalila García

Debido a que los datos de los agentes cambian constantemente, es necesario que el sistema que regula este tipo de cambios cumpla con dos características, es decir, que por una parte sea sencillo, y por otra que la respuesta sea la idónea. ¿Por qué es de carácter importante esto? Bien, porque el hecho de modificar el nombre de un contacto, en cierto modo puede demorarse, pero el hecho de modificar la potencia asociada a una unidad ofertante, es muy importante, por lo que la respuesta a la petición debe ser la correcta. Es por esto anterior, que el grupo de las tres preguntas siguientes pretende comprobar la calidad de esto.



The screenshot shows a web browser window with the address bar displaying 'IR_AQUI\CUESTIONARIO\DEFINITIVO_ENTREGAR'. The page title is 'ENCUESTA DE SATISFACION DE LOS AGENTES'. The page features a light blue header with a globe icon. The main content area is white and contains the following text:

ENCUESTA DE SATISFACION DE LOS AGENTES

SOBRE EL SISTEMA DE GESTION DE AGENTES

14. En caso de solicitar ayuda para modificar, actualizar o añadir datos referentes al domicilio social, denominación de su empresa etc... ¿La respuesta a sus solicitudes es la correcta?

- ☒ Siempre
- ☐ Casi Siempre
- ☐ Normalmente
- ☐ A veces
- ☐ Normalmente
- ☐ No aplica

15. En caso de necesitar ayuda o información para rectificar, actualizar o añadir datos relacionados a las garantías de pago ¿La respuesta a sus solicitudes es la correcta?

- ☐ Siempre
- ☐ Casi Siempre
- ☐ Normalmente
- ☐ A veces
- ☒ Normalmente
- ☐ No aplica

16. En caso de necesitar ayuda o información para rectificar, actualizar o añadir datos relacionados a las unidades físicas, unidades ofertantes entre otras, ¿La respuesta a sus solicitudes es la correcta?

- ☐ Siempre
- ☐ Casi Siempre
- ☐ Normalmente
- ☒ A veces
- ☐ Normalmente
- ☐ No aplica

At the bottom of the form, there are two buttons: 'Anterior' and 'Siguiente'. The name 'Dalila García' is visible at the bottom of the page.

El siguiente grupo de cuatro preguntas está relacionado con la usabilidad del sistema, es decir, básicamente si este es fácil de utilizar. Como se puede observar se insiste mucho en este aspecto, ya que es de gran importancia que el sistema sea sencillo, ya que de no ser así sería un gran impedimento para el conjunto de actuales y futuros agentes.



ENCUESTA DE SATISFACION DE LOS AGENTES

INDICADORES SUBJETIVOS

Las siguientes preguntas se realizan con el fin de evaluar la satisfacción de los agentes con la operativa que realiza el operador de mercado. Debe marcar una de las opciones que aparecen.

17. En caso de necesitar realizar una consulta en nuestra compañía, ¿Le resulta fácil contactar con la persona adecuada?

☒ Sí

☐ No

☐ No tengo respuesta

18. ¿Es fácil entender y rellenar las plantillas de datos que se usan en el web de agentes?

☐ Sí

☐ No

☒ No tengo respuesta

19. ¿Se ajusta el diseño de la web de agente a sus necesidades?

☐ Sí

☐ No

☐ No tengo respuesta

20. ¿Le resultó claro el proceso a seguir en nuestra compañía para adquirir la condición de agente?

☒ Sí

☐ No

☐ No tengo respuesta

[Anterior](#) [Siguiente](#)

Dalila García

Y por último nos encontramos con la última pantalla, cuyo principal objetivo es que el conjunto de agentes viertan opiniones o sugerencias que sean de importancia para ellos, con el fin de que la compañía, en caso de considerarlo oportuno tome las acciones necesarias para ponerlo en práctica.



ENCUESTA DE SATISFACION DE LOS AGENTES

COMENTARIOS

Si usted lo desea puede incluir en el cuadro adjunto sus comentarios, opiniones y sugerencias sobre el mercado y su operativa.

21. Sugerencias, comentarios u opiniones

Me gustaría que los precios del mercado diario llegaran por mensaje de texto a los agentes.

[Anterior](#)

[ENVIAR ENCUESTA](#)

Dalila García

15.5.3 Estudio de resultados obtenidos

Una vez que el conjunto de agentes ya han realizado la encuesta (en mi caso propongo que en mi sistema se ponga un mensaje con el enlace para acceder a esta, así como también la información acerca del plazo para realizarla), llega la hora de analizar las opiniones vertidas en ella.

En este caso, la idea no es obtener un resultado global de toda la encuesta, ya que para nuestro sistema en concreto no nos serviría de mucho. Quizá nos podamos preguntar ¿Qué hay de diferente en este sistema para que un resultado global no sea la mejor opción? Bueno, pues la respuesta es muy sencilla, un resultado global no es idóneo en este caso ya que el hecho de obtener un “sí, funciona correctamente”, un “no funciona correctamente” o un “funciona regular, hay que mejorar cosas”, no nos aporta información de gran importancia debido a que el sistema está estructurado en distintas secciones, es por esto, que la encuesta está organizada de la manera descrita en el apartado anterior.

Entonces, para sacar el mayor provecho de esta, los resultados se obtendrán por secciones. Recordamos que las secciones en las que está dividida la encuesta, son las siguientes:

- Datos generales del agente (número de años con condición de agente, potencia que negocia en los mercados, número medio de ofertas que presenta al día y número de UOF que dispone).
- Aspectos generales y de mayor peso de la operativa del mercado (horarios de apertura y cierre, retransmisión de información...).
- Sistemas de información y software (Accesos al sistema...).
- Liquidaciones y facturación (facturas en tiempo, fácil acceso a estas...).
- Gestión de agentes (relacionado con la modificación de datos relacionados con estos).
- Indicadores subjetivos (Facilidad de adquirir la condición de agente, facilidad para rellenar las plantillas necesarias...)
- Comentarios.

Continuando con el tema de análisis, llegados a este punto, se ha realizado una recolección de datos, los cuales es de vital importancia interpretarlos de una manera correcta para alcanzar el propósito que buscamos. Cada una de las opiniones vertidas en el cuestionario será minuciosamente analizada para ser presentadas de manera clara y concisa de forma que esto de lugar a una descripción real y de esta manera simplificar la planificación de acciones de mejora.

Como el cuestionario de satisfacción de los agentes está compuesto por preguntas cerradas, en mi opinión es recomendable usar estadísticos sencillos como son los gráficos, y tablas. Para comprender esto anterior mejor, a continuación realizaremos un ejemplo con la sección de indicadores objetivos:

	Siempre	Casi siempre	Normalmente	A veces	No aplica
¿Todas las sesiones comienzan en su debido tiempo?	78	65	15	3	1
¿Todas las sesiones finalizan en su debido tiempo?	78	65	15	3	1
En caso de notificar alguna noticia, ¿Dicha información es recibida a tiempo y en la forma correcta?	65	70	15	12	0
¿Los resultados de la casación se comunican de forma correcta?	70	65	20	7	0

Tabla 35: Resumen estadístico de las respuestas de los agentes

A continuación, el conjunto de datos, lo transformamos a un gráfico, ya que es mucho más sencillo de visualizar:

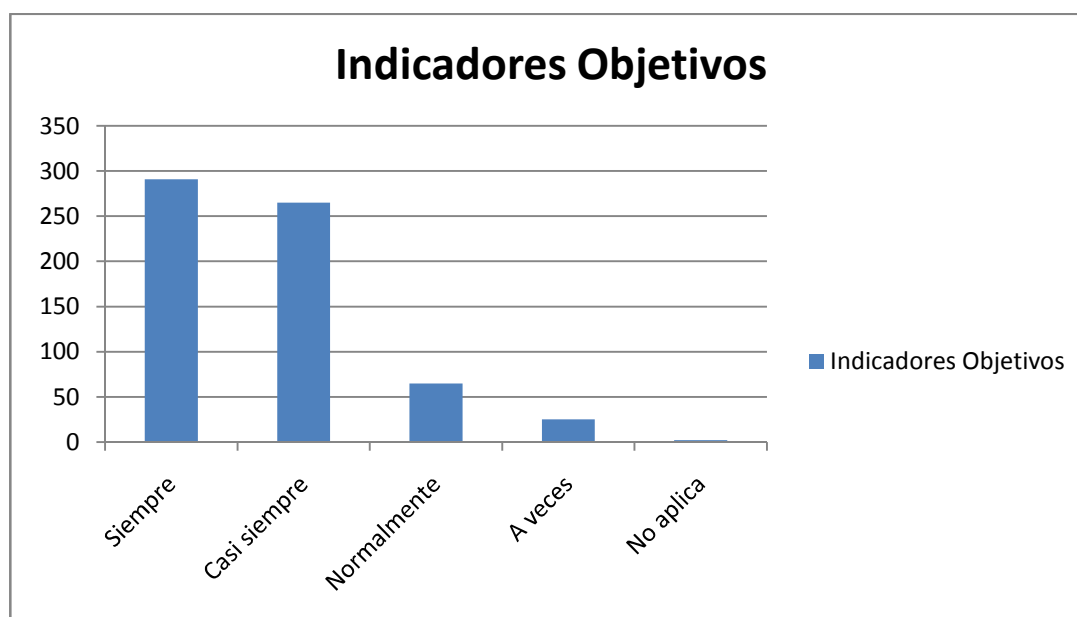


Ilustración 24: Gráfica correspondiente a los indicadores objetivos

Análisis de requisitos en el desarrollo del software

De la gráfica anterior, calculamos los porcentajes correspondientes a cada respuesta, ya que es mejor de visualizar:

	SIEMPRE	CASI SIEMPRE	NORMALMENTE	A VECES	NO APLICA
PORCENTAJE	44,9%	40,8%	10%	3,8 %	0,8%

Tabla 36: Porcentaje correspondiente a los indicadores objetivos

Por lo tanto, finalmente, teniendo ya los porcentajes disponibles, se puede analizar si hay que llevar a cabo acciones correctivas. Para ello, se busca el porcentaje obtenido en este caso en la opción “siempre” en la siguiente tabla, y en caso de necesitar estas, se llevan a cabo.

PORCENTAJE OBTENIDO	¿LLEVAR A CABO ACCION CORRECTIVA?
0% - 50%	Llevar a cabo acciones correctivas inmediatamente
51% - 60%	Llevar acciones correctivas en un corto periodo de tiempo
61% - 80%	Estudiar el caso, por si es necesario tomar acciones, aunque las probabilidades son remotas
81% - 100%	No realizar ninguna acción

Tabla 37: Relación entre porcentajes y acciones correctivas a tomar

Capítulo 13: presupuesto

13.1 Introducción

En este proyecto se ha realizado un análisis en profundidad sobre los requisitos en el desarrollo del software, así como también se han estudiado aspectos que derivan de estos como la calidad o estándares de calidad entre otros.

“El presupuesto total de este proyecto, asciende a la cantidad de 27.796,40 Euros”

Leganés a 05 de Octubre de 2015

El ingeniero proyectista

Fdo. Dalila García Notario

13.1.1 Planificación

Para realizar la planificación de este proyecto, ha sido necesario dividirlo en seis fases claramente diferenciadas:

- **FASE 1: Estudio Previo**

Donde se realiza la planificación de gran cantidad de información en relación al tema de requisitos software, así como también estándares (IEEE) y modelos (CMMI) vigentes que pudieran aportar información extra.

- **FASE 2: Proyectos software**

Una vez cogida una visión general de en qué consistía la fase de requisitos software, era necesario saber dónde y cuándo se llevaba a cabo esta dentro de un proyecto software, además de adquirir conocimientos acerca de estos.

- **FASE 3: Ingeniería de requisitos**

Debido a que se disponía de muchísima información acerca de requisitos software, toda ella proveniente de diferentes fuentes, pero principalmente de bibliografía adquirida en la biblioteca de la escuela politécnica de la universidad CarlosIII, se consideró de interés analizar la clasificación de la fase de ingeniería de requisitos según diferentes autores, con el fin entre otras cosas de observar similitudes y diferencias.

- **FASE 4: Requisitos**

Esta cuarta fase, fue la más extensa en tiempo, ya que engloba todas las actividades relacionadas con requisitos, es decir, desde las medidas de obtención de estos, tipos de requisitos, documento de especificación de requisitos hasta finalmente la validación de estos.

- **FASE 5: Desarrollo de requisitos adaptados al modelo CMMI**

Considere que debido a que el modelo CMMI es un conjunto de buenas prácticas y que muchas empresas de prestigio están evaluadas con este modelo como por ejemplo Accenture, AtosOrigin, IBM, Indra Software, sería interesante estudiar la gestión de requisitos software con dicho modelo y así así de esta manera aprender de ello.

- **FASE 6: Calidad de proyectos software**

Debido a que la calidad en proyectos software juega un papel muy importante, y también está ligado a una correcta especificación de requisitos, considere una buena opción para incluirlo. Es por esto que se incluye desde información general a herramientas y técnicas de calidad hasta normas y estándares.

13.1.2 Planificación del proyecto mediante diagrama de Gantt

A continuación se presenta un diagrama de Gantt desde se representa el conjunto de tareas y días empleados en cada una de ellas.

- Estudio previo: Del 1 de Septiembre al 13 de Octubre de 2014.
- Proyectos software: Del 14 de Octubre al 5 de Noviembre de 2014.
- Ingeniería de requisitos: Del 6 de Noviembre al 4 de Diciembre de 2014.
- Requisitos: Del 5 de Diciembre al 23 de Marzo de 2015.
- Desarrollo de requisitos adaptados al modelo CMMI: del 24 de Marzo al 4 de Mayo de 2015.
- Calidad de proyectos software 5 de Mayo al 24 de Agosto de 2015.

NOTA: en los periodos de tiempo anteriores están incluidas fechas de libranza en las que no se ha realizado proyecto como son una semana de navidad, una semana correspondiente a semana santa, una semana al mes de agosto y una semana en total de asuntos personales.

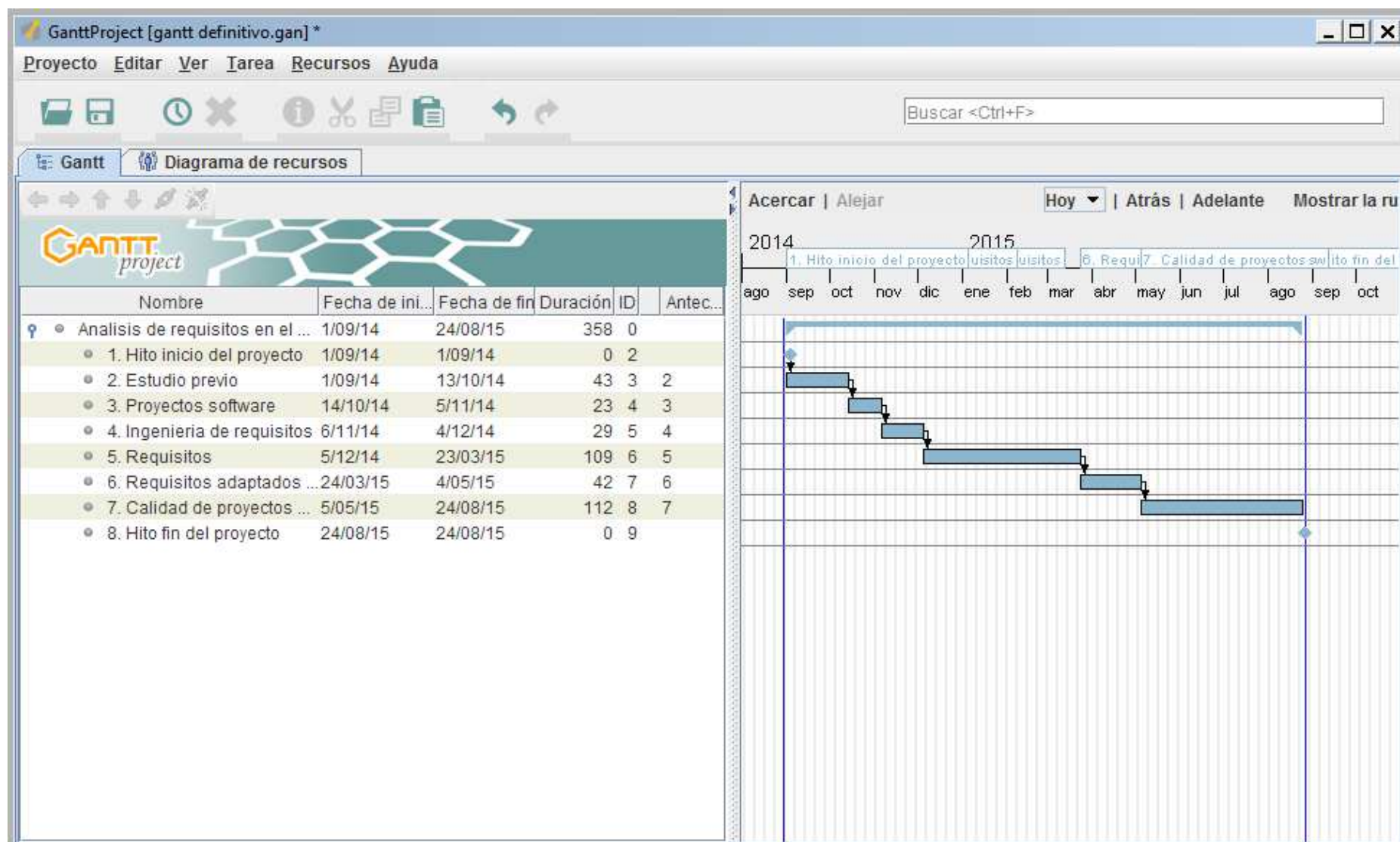


Ilustración 25: Diagrama de Gantt

13.1.3 Presupuesto

Para la realización del proyecto ha sido necesario el uso de unos recursos materiales mínimos, ya que aunque este se ha desarrollado a tiempo parcial (con una media de 4 horas al día), y con una única persona, es por esto que se ha empleado más tiempo, además también de contar con un presupuesto moderado.

1. **Autor:** Dalila García Notario
2. **Departamento:** Informática
3. **Descripción del proyecto:**
 - a. **Título:** Análisis de requisitos en el desarrollo del software
 - b. **Duración:** 12 meses
 - c. **Tasas de costes indirectos:** No aplica
4. **Presupuesto total del proyecto (valores en euros):** 27796,4Euros
5. **Desglose presupuestario (costes directos):**

PERSONAL

Apellidos y Nombre	N.I.F	Categoría	Dedicación	Coste persona mes	Coste (Euros)	Firma de conformidad
García Notario, Dalila		Ingeniero	10 personas/mes	2.694,39		
Personas mes			Total		26943,9	

Dice 10 personas/mes pero creo que sería mejor diez meses/persona

Duración del proyecto (quitando periodos donde no se realiza)= 1 de Septiembre de 2014 – 24 de Agosto de 2015 = 329 días * 4 horas/día (L-D) = 1316 horas.

Nota: en el cálculo de la duración del proyecto anterior, se han excluido un mes de trabajo correspondiente a una semana de navidad, una semana correspondiente a semana santa, una semana al mes de agosto y una semana de asuntos personales.

1316 horas / 131,25= 10,02 = 10 personas/mes

EQUIPOS

Descripción	Coste (euros)	% uso dedicado proyecto	Dedicación (Meses)	Periodo depreciación	Coste imputable
Software sin coste	885	100	12	60 meses	312,50

Análisis de requisitos en el desarrollo del software

Formula del cálculo de la amortización: $(A/B * C * D)$

A: número de meses desde la fecha de facturación en que el equipo es utilizado

B: Periodo depreciación (60 meses)

C: Coste equipo (Sin IVA)

D: % del uso que se dedicó al proyecto (habitualmente 100%)

A	B	C	D	TOTAL
25	60	750	100	312,50

SUBCONTRATACION DE TAREAS

No aplica.

OTROS COSTES DIRECTOS DEL PROYECTO

Descripción	Empresa	Coste imputable
Comida		350 euros
Transporte		190 Euros

6. Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	26943,9
Amortización	312,50
Subcontratación de tareas	0
Costes de funcionamiento	540
Costes indirectos	No aplica
Total	27796,4

Capítulo 14: Glosario

A

- **Accesibilidad:** La capacidad por la cual un software facilita su uso selectivo o el mantenimiento de sus componentes.
- **Adaptabilidad:** La facilidad con la que un software se ajusta a diferentes condicionantes del sistema y necesidades del usuario.
- **Análisis funcional:** Examen de una función definida con el fin de identificar todas las subfunciones necesarias para realizarla; identificación de las relaciones funcionales e interfaces (internas y externas) y captura de estas relaciones e interfaces en una arquitectura funcional; y descomposición de los requisitos de mayor nivel y asignación de estos requisitos a subfunciones de menor nivel.
- **Arquitectura funcional:** La organización jerárquica de las funciones, de sus interfaces funcionales internas y externas (externas a la propia agregación) e interfaces físicos externos, de sus respectivos requisitos, y de sus restricciones de diseño.
- **Aseguramiento de la calidad:** Conjunto de actividades diseñadas para evaluar el proceso mediante el cual los productos se fabrican.
- **Atributo de calidad:** Una propiedad de un producto o servicio por la que se juzgara su calidad por las partes interesadas relevantes. Los atributos de calidad son caracterizables por alguna medida adecuada.
- **Auditoría:** Es una revisión independiente cuyo objetivo es evaluar el cumplimiento de los requisitos software, instrucciones, especificaciones, códigos y normas establecidos, así como con respecto a los requisitos contractuales y a la efectividad de su implantación.

B

- **Base de datos (BBDD):** Conjunto de datos relacionados que van a ser utilizados por un grupo de usuarios o por un conjunto de programas.

C

- **Calidad del software:** (a) Conjunto global de características de un producto software, relacionado con su capacidad de satisfacer unas necesidades dadas. (b) Grado en el que un software posee una combinación de atributos deseada. (c) Grado en que el usuario percibe que el software satisface sus expectativas.

- **Caso de prueba:** Un conjunto específico de datos de prueba y los procedimientos asociados, desarrollados para un determinado caso, tal como probar un camino particular de un programa o verificar el cumplimiento de un requisito específico.
- **Ciclo de vida del software:** Periodo de tiempo que comienza cuando se concibe un producto software y finaliza cuando el producto pierde su utilidad. El ciclo de vida del software incluye las siguientes fases: fase de requisitos, fase de diseño, fase de realización, fase de pruebas, fase de instalación y aceptación, fase de operación y mantenimiento y algunas veces fase de retirada.
- **Clases de análisis:** entidades del dominio del negocio visibles para el usuario final.
- **CMMI:** Enfoque integrado para el modelado de madurez de la capacidad del proceso. Apoya los modelados de madurez discretos y continuos e integra sistemas y modelos de madurez de los procesos de la ingeniería del software.
- **Comando:** Una expresión que puede ser introducida en un sistema informático para iniciar una acción o afectar a la ejecución de un programa de ordenador; por ejemplo, el "inicio de sesión" comando para iniciar una sesión de ordenador.
- **Compilador:** Programa usado para compilar.
- **Compilar:** Traducir un programa en lenguaje de alto nivel a su equivalente en código máquina absoluto o reubicable.
- **Complejidad:** Grado de complicación de un sistema o componente de un sistema, determinado por factores tales como el número e interrelación de las interfaces y ramas condicionales, el grado de anidamiento, los tipos de estructuras de datos y otras características del sistema.
- **Componente:** Una de las partes que componen un sistema. Un componente puede ser hardware o software y puede ser subdividido en otros componentes.
- **Componente de producto:** Un producto de trabajo que es un componente de bajo nivel del producto.
- **Conexión:** Asociación establecida entre unidades funcionales para transmitir información.
- **Configuración:** (a) Disposición de un sistema de ordenadores o red definida por el número, naturaleza y principales características de sus unidades funcionales. Más específicamente, el término configuración puede referirse a configuración hardware o configuración software. (b) Requisitos, diseño y realización que definen una versión particular de un sistema o componente. (c) Características físicas o funcionales del hardware o software establecidas en la documentación técnica y realizada en un producto.

- **Consistencia:** La consistencia significa que los requisitos no deben tener definiciones contradictorias.
- **Control de cambios:** El proceso por el que un cambio se propone, evalúa, aprueba o rechaza, programa y controla su introducción.

D

- **Datos:** Una representación de los hechos, conceptos o instrucciones de una manera adecuada para la comunicación, interpretación o proceso por medios humanos o automáticos.
- **Datos de prueba:** Datos desarrollados para probar un sistema o componente de un sistema.
- **Desarrollo:** (a) Realización de una abstracción en términos más concretos. En particular, en términos de hardware, software o ambos. (b) Forma ejecutable en máquina de un programa, o forma de un programa que puede ser traducido automáticamente a una forma ejecutable en máquina. (c) Proceso de codificar un diseño y de depurar el código.
- **Descendente:** Relativo a un método que comienza con los componentes de más alto nivel jerárquico y actúa a través de niveles cada vez más bajos.
- **Diseño:** El proceso de definición de la arquitectura software, componentes, módulos, interfaces, procedimientos de prueba y datos de un sistema software para satisfacer unos requisitos especificados.
- **Disponibilidad:** (a) La probabilidad de que el software sea capaz de realizar su función cuando se necesite. (b) La capacidad de que un elemento cumpla con su función cuando se requiera su uso.

E

- **Eficiencia:** El grado en que un sistema o componente realiza sus funciones designadas con un consumo mínimo de recursos.
- **Entidad:** En la programación de computadoras, cualquier elemento que puede ser nombrado en un programa. Por ejemplo, un elemento de datos.
- **Entrada:** (1) Perteneciente a los datos recibidos desde una fuente externa. (2) Perteneciente a un dispositivo, proceso o canal implicado en la recepción de datos desde una fuente externa. (3) Para recibir datos desde una fuente externa. (4) Para proporcionar los datos de una fuente externa.
- **Especificación de requisitos software (ERS):** Documento que se crea cuando debe especificarse una descripción detallada de todos los aspectos del software que se va a

elaborar, antes de que el proyecto comience. Es importante notar que una ERS formal no siempre está en forma escrita.

- **Error:** (1) Diferencia entre un valor computado, observado o medido y el valor verdadero, determinado o teóricamente correcto. (2) Un paso incorrecto, proceso o definición de datos. (3) Un resultado incorrecto. (4) A la acción humana que produce un resultado incorrecto
- **Estándar:** Requisitos obligatorios empleados y aplicados a prescribir un enfoque uniforme disciplinado para el desarrollo de software, es decir, las convenciones y prácticas obligatorias están en los estándares.
- **Estructura de datos:** Una representación formalizada de las relaciones de orden y la accesibilidad entre los datos, sin tener en cuenta su configuración real en memoria.

F

- **Fallo:** (a) La pérdida de la capacidad de una unidad funcional para realizar su función requerida. (b) La incapacidad de un sistema o componente de un sistema para realizar una función requerida dentro de límites especificados. Puede producirse un fallo cuando se manifiesta un defecto. (c) Desviación entre la operación de un programa y sus requisitos.
- **Fase de desarrollo:** Periodo de tiempo en el ciclo de vida del software durante el cual se crea un producto software a partir de la documentación de diseño y se depura.
- **Fase de requisitos:** El periodo de tiempo durante el ciclo de vida del software en el que se definen y documentan los requisitos para un producto software, tales como las capacidades funcionales y prestaciones.
- **Flexibilidad:** La facilidad con la que un sistema o componente puede ser modificado para su uso en aplicaciones o entornos distintos de aquellos para los que fue diseñado específicamente.

G

- **Gestión de la calidad:** Manera en la que se gestiona una organización para lograr la excelencia. Actividades coordinadas para dirigir y controlar una unidad en lo relativo a la calidad. La gestión de la calidad incluye: la prefijación estratégica, la asignación de recursos y otras actividades sistemáticas, tales como la planificación, las operaciones y las evaluaciones relativas a la calidad.

I

- **Identificador:** El nombre, la dirección, o el índice que identifica un objeto en un programa informático.

- **Integridad:** El grado en que un sistema o componente impide el acceso no autorizado a, o modificación de programas de ordenador o de datos.
- **Interactivo:** Perteneciente a un sistema o modo de funcionamiento en el que cada entrada del usuario provoca una respuesta de acción o por el sistema.
- **Interfaz:** Especificación de los atributos y operaciones asociados con un componente software. La interfaz es utilizada como el medio de tener acceso a la funcionalidad del componente.
- **ISO 9000:** Acrónimo de Internacional Standards Organization. Conjunto de normas que certifican que una organización dispone de un sistema de calidad acorde a una serie de requisitos y recomendaciones definidas por la misma.

L

- **Lenguaje:** (1) Medio sistemático de la comunicación de ideas mediante el uso de signos, sonidos, gestos o signos y reglas para la formación de expresiones admisibles. (2) (IEEE Std 830-1984 [61]) Un medio de comunicación, con la sintaxis y la semántica, que consiste en un conjunto de representaciones, convenciones y reglas asociadas que se utilizan para transmitir información.
- **Lenguaje natural:** Un lenguaje cuyas reglas se basan en el uso en lugar de ser preestablecido antes del uso de la lengua.
- **Línea base de requisitos:** Según el estándar IEEE, se define como una especificación o producto que ha sido revisada formalmente y sobre la que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios.
- **Lluvia de ideas:** Método para alentar el razonamiento creativo, mediante juicios diferidos. El propósito del juicio diferido es alentar a las personas a aportar sugerencias osadas y singulares, sin preocuparse sobre lo que los demás piensen acerca de ellas.

M

- **Mantenimiento del software:** Modificación de un producto software después de su entrega, para la corrección de defectos, para mejorar sus prestaciones u otros atributos o para adaptar el producto a un entorno diferente.

O

- **Objeto:** (1) Perteneciente a los resultados de un proceso de ensamblaje o compilación. (2) Una constante de programa o variable. (3) Una encapsulación de datos y servicios que manipulan los datos.

P

- **Pan de desarrollo software:** Un plan del proyecto para el desarrollo de un producto software.
- **Plan de proyecto:** Un documento de gestión que describe la realización del proyecto. El plan describe, entre otros, el trabajo que ha de llevarse a cabo, los recursos necesarios y los métodos a utilizar.
- **Producto:** Un producto de trabajo que está previsto entregar a un cliente o usuario final.
- **Producto de trabajo:** Un resultado útil de un proceso.
- **Programa:** Conjunto de instrucciones adecuadas para ser procesadas en un ordenador. El proceso puede incluir el uso de un ensamblador, compilador, intérprete o traductor, tanto para preparar la ejecución del programa como para ejecutarlo.
- **Protocolo:** conjunto de reglas que gobiernan la operación de unidades funcionales para llevar a cabo una comunicación.
- **Prototipo:** Sistema software que incluye solo algunas características del sistema final con objeto de mejorar la comprensión del diseño o de su realización.
- **Prueba:** Proceso de ejecutar o evaluar un sistema o componente de un sistema por medios manuales o automáticos, para comprobar que cumple los requisitos especificados, o para identificar las diferencias entre los resultados esperados y los obtenidos.
- **Prueba del sistema:** El proceso de prueba de un sistema integrado hardware o software, para verificar que el sistema cumple los requisitos especificados.

R

- **Recuperación de fallos:** el retorno de un sistema a un estado fiable de operación después de un fallo.
- **Requisito:** Una condición o característica que debe tener o cumplir un sistema o componente de un sistema para satisfacer un contrato, norma, especificación, u otro documento formalmente impuesto. El conjunto de todos los requisitos forma la base para el desarrollo consiguiente del sistema o componente del sistema.

S

- **Salida:** (1) Perteneciente a los datos transmitidos a un destino externo. (2) Perteneciente a un dispositivo, proceso o canal involucrado en la transmisión de datos a un destino externo. (3) Para transmitir datos a un destino externo.
- **Simulación:** Un modelo que se comporta o funciona como un sistema dado cuando se proporciona un conjunto de entradas controladas.
- **Sistema operativo:** software que controla la ejecución de programas. Un sistema operativo puede proporcionar servicios tales como asignación de recursos, control de entrada/salida y gestión de datos. Aunque los sistemas operativos son predominantemente software, son posibles implantaciones hardware parciales o totales.
- **Software:** (a) Programas, procedimientos, reglas y la posible documentación asociada y datos que pertenezcan a la explotación de un sistema de ordenador. (b) Programas, procedimientos, reglas o cualquier documentación asociada que pertenezcan a la explotación de un ordenador.
- **Stakeholders:** cualquier persona o grupo que se verá afectado por el sistema, directa o indirectamente. Entre los stakeholders se encuentran los usuarios finales que interactúan con el sistema y todos aquellos en la organización que se pueden ver afectados por su instalación.

T

- **Tipo de datos:** Una clase de datos caracterizada por los miembros de la clase y las operaciones que se les pueden aplicar.
- **Trazabilidad:** La trazabilidad es la habilidad para seguir la vida de un requerimiento en ambos sentidos, hacia sus orígenes o hacia su implementación a través de las especificaciones generadas durante el proceso de desarrollo. Es un factor de calidad.

U

- **Unidad funcional:** Entidad de hardware, software o de ambas, capaz de cumplir un propósito específico.
- **Usabilidad:** La medida con la que un producto se puede usar por usuarios determinados para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso concreto.

V

- **Validación:** Proceso de evaluar el software al final del proceso de desarrollo de software para asegurar el cumplimiento de los requisitos.

- **Verificación:** (a) Proceso para determinar si los productos de una fase determinada del ciclo de desarrollo del software cumplen o no los requisitos establecidos durante la fase previa. (b) Prueba formal de la exactitud de un programa. (c) Actividad de revisar, inspeccionar, probar, comprobar, auditar o cualquier otra acción que establezca y documente si los elementos, procesos, servicios, o documentos están o no de acuerdo a los requisitos especificados.

Capítulo 15: Siglas

[AENOR]	Asociación Española de Normalización y Certificación
[AMFE]	Análisis Modal de Fallos y Efectos
[ANSI]	American National Standards Institute
[CMMI]	Capability Maturity Model Integration
[COQ]	Coste de la calidad
[CPD]	Centro de procesamiento de datos.
[DOE]	Design Of Experiments
[DRAE]	Diccionario Real Academia Española
[EFQM]	European Foundation for Quality Management
[ERS]	Especificación de Requisitos Software
[FTP]	File Transfer Protocol
[ISO]	International Organization for Standardization
[MD]	Mercado Diario
[OWS]	Object Breakdown Structure
[PCS]	Plan de Calidad Software
[PDS]	Plan de Desarrollo Software
[PMBOK]	Project Management Body of Knowledge
[PX]	Power Exchange
[QFD]	Quality Function Deployment
[RBS]	Resource Breakdown Structure
[SRS]	Software Requirements Specification
[UNE]	Una Norma Española
[UOF]	Unidad Ofertante
[WBS]	Work Breakdown Structure

Capítulo 16: Bibliografía

LIBROS

Braude, Eric J.: 'Ingeniería de Software: Una Perspectiva orientada a objetos'. RA-MA Editorial, Madrid 2003. [Bra03]

Comité de calidad del Software de la A.E.C.C.: "Glosario de Términos de calidad e ingeniería del Software".

Cuevas Agustín, Gonzalo.: "Gestión del proceso software". EDITORIAL CENTRO DE ESTUDIOS RAMON ARECES, S.A. Madrid, 2002. [Cue02]

Dorfman, Merlin; Thayer, Richard H.: "Software Requirements engineering". Second Edition. IEEE Computer Society Press. 1997. [DT97]

Lauesen, Soren.: "Software requirements: Styles and Techniques". ADDISON-WESLEY, Britain 2002. [Lau02]

Lopez-Cortijo, Román; de Amescua Seco, García Antonio.: "Ingeniería del software: Aspectos de gestión. Tomo 1- Conceptos básicos. Teoría, Ejercicios y herramientas". IIS (Instituto Ibérico de la Industria del Software). Madrid, 1998. [LA98]

PiattiniVelthius, Mario G.; García Rubio, Félix O.; Caballero Muñoz-Reja, Ismael.: "Calidad de sistemas informáticos". RA-MA editorial. Madrid, 2006. [PGC06]

Sommerville, Ian.: 'Ingeniería del Software'. 7ª Edición. PEARSON EDUCACIÓN, S.A. Madrid 2005. [Som05]

Somerville, Ian; Sawyer, Pete.: 'A good practice guide'. WILEY. London, 1997. [SS97]

S. Pressman, Roger.: IN 'Ingeniería del Software': Un enfoque práctico. 7ª Edición. McGraw Hill. 2010. [Pre10]

Wiegers, Karl E.: "Software requirements: Practical techniques for gathering and managing requirements throughout the product development cycle". Microsoft Press. 2003. [Wie03]

NORMAS

IEEEStd 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology. 84 páginas.

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. 37 páginas.

IEEE Std 830-1984 IEEE Guide for Software Requirements Specifications. 26 páginas.

ISO/IEC 25030 INTERNATIONAL STANDARD Software engineering – Software product Quality Requirements and Evaluation. 36 páginas.

GUIAS

A guide to the Project Management Body of Knowledge (PMBOK Guide). Diciembre, 2012. 211 páginas.

CMMI® para Desarrollo, Versión 1.3. Software Engineering Institute. Noviembre, 2010. 555 páginas.

PÁGINAS O DOCUMENTOS ELECTRONICOS EN LA RED

<http://compromisosdecalidad.es/wiki/index.php/Glosario>

http://www.famp.es/racs/ObsCultura/Docs/guia_sccc/9_Glosario.pdf

<http://www.pmigdl.org/?p=949>

<http://www.projectsmart.co.uk/pdf/the-curious-case-of-the-chaos-report-2009.pdf>

http://www.codecompiling.net/files/slides/UML_clase_02_UML_casos_de_uso.pdf

http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf

Capítulo 17: Conclusiones

Debido a los problemas ocasionados y a las deficiencias existentes actualmente en las empresas por una malarealización de la especificación de requisitos software, dando mayor importancia a aspectos insignificantes, en mi opinión el proyecto aporta a las empresas al menos el grado de importancia de esto con el fin de evitar o subsanar problemas repetitivos en el trabajo diario, ya que directamente, la especificación de requisitos está ligada con la calidad, y por consiguiente, esta última es la principal fuente de supervivencia de las compañías actualmente.

17.1 Vías de investigación futuras

Una vez realizado este proyecto se propone lo siguiente como línea de investigación futura:

- Herramienta cuya principal funcionalidad sea la de verificar si la especificación de requisitos software realizada para un determinado proyecto es correcta, todo ello con el fin de obtener mejores resultados. Dicha herramienta debe poseer una serie de características, como son las siguientes:
 - Sencilla para que todos los usuarios puedan utilizarla
 - Útil, es decir, que presente una alta capacidad de análisis.
 - Usable, esto es, que esta pueda ser utilizada en todos los niveles de la organización.
 - Fiable, esto es, que los resultados devueltos por la herramienta, sean lógicos.

17.2 Opiniones personales

Mediante la realización de este proyecto final de carrera he tenido la oportunidad de reforzar los conocimientos ya adquiridos en la universidad, durante los años de la carrera, y por otra parte me ha venido bien ya que profesionalmente, después de realizar este, a día de hoy comprendo a la perfección la dinámica que se sigue en un proyecto, ya sea nuevo o una ampliación del ya existente cuando se trata de temas relacionados con los requisitos software.

Muchos de los aspectos que se tratan en este proyecto, prácticamente eran desconocidos para mí, pero a raíz de profundizar en cada uno de los temas, e invertir muchas horas de trabajo en cada uno de ellos, he tenido la oportunidad de analizar y comprender la importancia de cada uno de ellos.

Como conclusión, me gustaría indicar que para mí, este proyecto, a la vez de divertido ya que la ingeniería del software así como la calidad, son temas que me apasionan, ha sido muy productivo, ya que he llegado a comprender la gran importancia que tiene el realizar una especificación de requisitos adecuada así como emplear el tiempo necesario en ella, de cara a evitar graves repercusiones.

Capítulo 18: Anexo

18.1 Introducción caso práctico

Existen dos tipos de mercados a los que nosotros nos estaremos refiriendo a lo largo de este caso práctico: VALE, PERO QUIZÁ ESTA EXPLICACIÓN AL PRINCIPIO, O QUE AL MENOS PARA QUE LO ENTIENDAN MEJOR

- **Mercado Diario (MD)**: Es un mercado donde se llevan a cabo las operaciones de compra y venta de energía para el día siguiente (D+1). Por ejemplo, hoy día 12/7/2015 (día D) se casará la energía para las 24 horas del 13/07/2015 (día D+1). Por lo tanto, el conjunto de agentes pueden presentar ofertas hasta las 12:00 horas del día D.
- **Mercado Intradiario (MI)**: Son también a los que llamamos “mercados de ajuste” y se realizan seis mercado a lo largo del día con sus horizontes horarios correspondientes. El principal objetivo de estos mercados es ajustar las cantidades adquiridas en el mercado diario con posterioridad a haberse fijado este.

Mercado	Horizonte horario
Intradiario 1	17:00 – 18:45
Intradiario 2	21:00- 21:45
Intradiario 3	01:00 – 01:45
Intradiario 4	04:00 – 04:45
Intradiario 5	08:00 – 08:45
Intradiario 6	12:00 – 12:45

Con el fin de poner en práctica los aspectos estudiados en el proyecto, a continuación se procede a realizar un caso práctico relacionado con el “El mercado eléctrico”. He optado por este camino por una parte porque desde mi punto de vista es un tema muy interesante, debido a que la mayoría de la gente no tiene ningún conocimiento sobre ello y por otro porque es a lo que me dedico desde hace dos años y medio, por lo que mi conocimiento sobre dicho tema cada vez es más amplio.

Debido a que nos encontramos con un sistema que es poco común y poco conocido, y con un cierto grado de dificultad, a continuación se aporta más cantidad de información general referente a algunos conceptos que es necesario saber, debido a que estos serán utilizados a lo largo del caso práctico:

- Con el fin de poder operar en el mercado de la electricidad, es requisito imprescindible darse de alta como **AGENTE**, pero ¿Quiénes son los agentes? ¿Qué funciones realizan estos? Pues bien, según lo que establece el Boletín del Estado (número 11 / Sábado 10 de Mayo de 2014), los agentes serán productores, consumidores, comercializadores, representantes y gestores de cargas del sistema, los cuales participan en el mercado diario y mercado Intradiario presentando ofertas de compra y venta con el fin de adquirir/vender energía.

Con el fin de conocer que hacen cada uno de los sujetos de mercado mencionados anteriormente, a continuación se aporta una pequeña definición de cada uno de ellos, tomadas del Boletín del Estado correspondiente:

Sujeto de mercado	Definición
Productor	Persona física o jurídica que tiene la función de generar energía eléctrica, así como también la de construir, operar y mantener las instalaciones de producción.
Consumidor	Es la persona física o jurídica que compra energía para su propio consumo.
Comercializador	Persona jurídica que accediendo a las redes de transporte o distribución tienen como función la venta de energía eléctrica a los consumidores o a otros sujetos del sistema.
Representante	Son aquellos que actúan por cuenta de un sujeto del mercado, ya sea en nombre de dicho sujeto, en ese caso estaríamos hablando de representación directa o ya sea en nombre propio lo cual sería una representación indirecta.
Gestores de cargas del sistema	Se refiere a sociedades mercantiles de servicios de recarga energética.

- **PX** (Power Exchange): se refiere al operador de mercado de energía eléctrica correspondiente a cada zona europea, esto es, España tiene un Px que somos nosotros, Italia tiene otro Px, y así sucesivamente.
- **TSO**: en nuestro caso en particular, red eléctrica es nuestro TSO, y dentro del mercado eléctrico español se corresponde con nuestro operador del sistema eléctrico. ¿Cuál es su función? Principalmente sus funciones son las siguientes:
 - Asegurar que el funcionamiento en este caso del sistema eléctrico es el adecuado.
 - Verificar que el suministro de energía eléctrica es continuo y seguro.

Llegados a este punto, ya tenemos una primera visión, por lo tanto el caso práctico consistirá en lo siguiente:

“Propondré el conjunto de requisitos de mayor importancia necesarios para los diferentes sistemas utilizados para la casación del mercado diario a nivel europeo. He elegido esta opción, ya que desde mi punto de vista, y basta que conozco en profundidad el tema, la casación diaria podría ser realizada de una forma más simplificada de la existente actualmente”.

A modo informativo, actualmente se realiza la casación del mercado diario, como su nombre indica todos los días a las 12:00 y esta es coordinada por cinco PX's. Debido a que no pueden coordinar todos a un mismo tiempo el mercado eléctrico, siempre existe una figura de coordinador y otro Px con rol de backupcoordinador. Este rol va cambiando de Px cada 15 días. Si bien es cierto, aunque como hemos mencionado existe una figura de coordinador y de backupcoordinador, en la actualidad es imprescindible la cooperación de todos los Px's para poderse llevar a cabo la casación del mercado.

El procedimiento a seguir que propongo para realizar dicha casación del mercado diario es la siguiente, y la cual en procedimiento es muy diferente a la actual:

- Existirá un Px (Power Exchange) principal, en este caso el español, a lo largo del caso práctico lo nombraremos como Px coordinador, el cual realizará la casación completa, sin necesidad

de la necesidad de cooperación del resto de Px's. Adicionalmente, los cinco Px's restantes podrán realizar un seguimiento de la casación con el fin de poder informar a sus agentes en caso de problemas.

¿Cómo se llevará a cabo esto?

SITUACION ACTUAL	SITUACION PROPUESTA
Cada TSO (en nuestro caso REE), envía su fichero de capacidad a su correspondiente Px a través de un FTP. Por lo que cada Px debe enviar su fichero de capacidad al Px coordinador a través de la nube.	Propongo que cada TSO envíe su fichero de capacidad al Px coordinador a través de un FTP compartido entre el Px y el conjunto de TSO's. En este caso, dicho Px debe subir a una determinada plataforma (que se habla de ella en los pasos siguientes de ella) cada uno de los ficheros. Cada uno de estos ficheros llevará asociado el nombre referente a la zona que representan.
El conjunto de agentes envía sus ofertas al PX que le corresponde, para posteriormente dicho PX enviar el fichero de ofertas al coordinador a través de la nube una vez cerrado el mercado.	Propongo que el conjunto de agentes existentes de todos los PX's envíen sus ofertas al Px coordinador, en este caso al español. Una vez cerrado el mercado, a las 12:00 horas, el PX realiza una preparación de ofertas global obteniendo el fichero de ofertas correspondiente, el cual lo debe de subir a una determinada plataforma.
Una vez que la plataforma dispone de todos los ficheros de capacidad, y todos los ficheros de ofertas, empieza el cálculo en el que como salida de este obtendremos el precio resultante de la casación.	En este caso, no tendremos problema, ya que al no existir intermediarios, y ser un único Px el que dispone de todos los datos, no se puede dar el caso de retrasos debido a que no se disponga de todos los ficheros.
Una vez que ya se ha realizado la casación, los resultados deben distribuirse a cada Px para que estos se publiquen a los agentes correspondientes. Esto conlleva un tiempo además de asumir ciertos riesgos, ya que en el caso de que alguna de las conexiones con algún Px falle, ya hay que tomar medidas de backup.	Aquí, debido a que todos los agentes se conectan a nuestra plataforma, se ahorra tiempo debido que nosotros directamente publicamos a los agentes. A parte de ahorrar tiempo, también ahorramos los problemas mencionados en la situación actual.
Cada Px debe enviar al coordinador, a través de la nube la confirmación de resultados.	En este caso, directamente, en caso de no existir reclamaciones por parte de los agentes, el único PX sube a la plataforma la confirmación de los resultados.

18.2 Sistemas necesarios

Para llevar a cabo la casación del Mercado diario a nivel europeo, se necesitarán tres sistemas donde deben acceder los agentes y operadores del sistema. A continuación se adjunta un cuadro que nos informa para cada uno de los sistemas porque usuarios es utilizado, y algunas de las principales funciones que aporta dicho sistema.

NOMBRE DEL SISTEMA	¿QUIEN LO UTILIZA?	PRINCIPALES FUNCIONES REALIZADAS CON ESE SISTEMA
Web principal	Agentes	<ul style="list-style-type: none">- Insertar ofertas para el mercado diario- Consultar resultados del MD (precios, energías casadas...)- Consultar información relacionada con liquidaciones (cantidad a pagar, día de pago, horizontes horarios de solicitudes...)- Consultar la fase en la que se encuentra el mercado (recepción de ofertas, casación, recepción de datos...)- Realizar peticiones (modificación de datos de contacto, modificación de la potencia de una UOF...)
	Operadores / PX coordinador	<ul style="list-style-type: none">- Consultar datos específicos de una oferta del MD- Consultar si existen reclamaciones insertadas por los agentes en el plazo estipulado.- Consultar información general (datos de contacto, resultados del mercado, garantías disponibles...)
Plataforma de operación	Operadores / PX coordinador	<ul style="list-style-type: none">- Realizar la casación del MD (preparación y envío del fichero correspondiente)- Realizar el envío de confirmaciones, subir ficheros así como también visualizar el precio de la casación.
Plataforma de acceso a la nube	Operadores / PX coordinador	<ul style="list-style-type: none">- Subir fichero de capacidad y confirmación correspondientes a todos los Px's- Manipular el sistema en general para que el resto de Px puedan realizar el seguimiento.
	Resto de Px's	<ul style="list-style-type: none">- Realizar seguimiento de la casación- Posibilidad de descargarse los ficheros del resto de Px's (capacidades, confirmaciones...)- Consultar el precio resultante de la casación.

Adicionalmente a esto, debe ser necesario utilizar una plataforma de visualización (disponible ya en el mercado), con el fin de que el Px coordinador comparta su pantalla todos los días, con el fin de que el resto de Px puedan visualizar los pasos seguidos por este.

Una vez llegados a este punto y conocidos los sistemas que se deben utilizar así como también el supuesto nuevo funcionamiento para la casación del mercado diario, probablemente nos podremos preguntar: si toda la operativa la lleva a cabo un Px, ¿Qué función desempeñan el resto? Pues esta pregunta es muy fácil de responder, es de vital necesidad que el resto de Px's estén activos y con sus sistemas preparados, ya que en caso de problemas, cada PX deberá realizar la casación del mercado diario con sus sistemas de backup.

18.3 Análisis de los diferentes sistemas

18.3.1 Actores del sistema

En un primer lugar, debido a que se trata de una aplicación privada, de uso exclusivo para determinados usuarios, y en particular para un conjunto de agentes que compran y venden energía, y un conjunto de operadores, Px's y supervisores que controlan cada uno de los diferentes mercados, los principales roles existentes que interactúan con dicha aplicación, se detallan a continuación:

LISTADO DE ACTORES		
IDENTIFICADOR	NOMBRE	DESCRIPCION
AC-1	Agente	Persona física cuya principal función consiste en insertar ofertas de compra o venta para cada uno de los seis mercados que se realizan a lo largo del día. Además dispondrá de toda la información relativa a cada uno de estos en el “Web de agentes”.
AC-2	Operador/Px/supervisor	Persona física encargada de interactuar con la aplicación durante los diferentes mercados, con el fin principal de obtener un precio horario para cada hora que se negocia, así como también para consultar información relativa a cada uno de los mercados y hacer el oportuno seguimiento de estos.
AC-4	Ingeniero de pruebas	Persona física encargada de realizar todas las pruebas pertinentes y necesarias para verificar el correcto funcionamiento de la aplicación, así como también comprobar que dicha aplicación contiene el conjunto de funcionalidades necesarias para llevar a cabo la tarea para la que ha sido realizada.

Una vez que ya sabemos el conjunto de funcionalidades que debe prestar cada uno de los sistemas que van a ser utilizados, debemos de sacar el conjunto de requisitos necesarios y de mayor importancia para cubrir las necesidades anteriores, por lo que a continuación se describen estos. Si bien es cierto, solo realizaremos una especificación de requisitos funcionales y requisitos no funcionales ya que estos son los que están directamente relacionados con el sistema.

18.3.2 Requisitos funcionales

REQUISITOS FUNCIONALES				
IDENTIFICADOR	ACCION	ACTOR	SISTEMA QUE DEBE SER UTILIZADO	DESCRIPCION
RF-01	Insertar una oferta de compra/venta	Agente	Web principal	Permite a todos los agentes realizar una determinada oferta.
RF-02	Consulta de reclamaciones de agentes	Operador/Px	Web principal	Permite consultar el conjunto de reclamaciones realizadas por los agentes para cada uno de los mercados.
RF-03	Consultar datos de ofertas realizadas	Agente / Operador / Px	Web principal	La aplicación debe permitir consultar a los diferentes usuarios del sistema información relativa

				acerca de las ofertas insertadas para el mercado diario.
RF-04	Visualizar la información general de los mercados próximos	Agente / Operador / Px	Web principal	El sistema debe mostrar la principal información de los mercados próximos.
RF-05	Consulta de transacciones firmadas	Operador / Px	Web principal	Consulta de todas las transacciones firmadas.
RF-06	Acceso al web principal	Agente / Operador / Px/ ingeniero de pruebas	Web principal	El sistema debe permitir acceder a ella a los usuarios de la misma.
RF-07	Acceso a la plataforma de operación	Operador / Px/ ingeniero de pruebas	Plataforma de operación	El sistema debe permitir acceder a ella a los usuarios de la misma.
RF-08	Navegar utilizando el menú superior de las aplicaciones.	Agente / Operador / Px/ ingeniero de pruebas	Web principal/ Plataforma de operación/ Plataforma de acceso a la nube	Los sistemas permitirán navegar por los diferentes submenús para consultar la información necesaria.
Rf-09	Subir fichero de capacidad a la nube mediante la plataforma de acceso a la nube	Operador / Px coordinador	Plataforma de acceso a la nube	La plataforma de acceso a la nube debe permitir subir a la nube los ficheros de capacidad.
RF-10	Realizar una reclamación	Agente	Web principal	El sistema debe permitir al agente poner una reclamación al mercado.
RF-11	Recibir los agentes un mensaje de texto con los resultados del precio del mercado	Agente	Plataforma de operación	Desde la plataforma de operación se debe poder enviar diariamente unSMS con el precio resultante.

LISTADO DE REQUISITOS DETALLADOS	
IDENTIFICADOR	DESCRIPCION
RF-01	El sistema debe permitir a una persona física asignada con condición de agente, insertar un número limitado de ofertas para el mercado diario dentro del horizonte horario permitido. Para ello, el web de agentes dispone de diferentes pestañas, cada una de ellas correspondiente a un tipo de mercado, donde podrán insertar la oferta por pantalla o bien por fichero.
RF-02	El sistema debe permitir consultar las reclamaciones expuestas por los agentes para un determinado mercado. Para ello el sistema debe proporcionar una opción donde te dé a elegir el mercado que deseas consultar (diario e Intradía) y número de sesión en caso de tratarse del mercado Intradía y como respuesta se debe obtener el total de reclamaciones existentes, en caso de haber alguna.
RF-03	El sistema debe permitir consultar tanto a agentes, operadores y Px coordinador la información relativa a las ofertas presentadas al mercado diario. Entre otra, la información que deben poder consultar es el conjunto de valores que identifican la oferta, datos generales de esta, condiciones económicas y condiciones técnicas exigidas por el ofertante y el detalle de la oferta que ha sido presentada, es decir, la cantidad de energía que se quiere comprar y vender en cada hora y el precio de la misma por Mwh.
RF-04	El sistema debe mostrar la principal información de los mercados próximos y dicha información debe ser correcta y actualizada. La principal información que se debe poder visualizar es la siguiente: <ul style="list-style-type: none"> • Información del mercado diario: día y fecha de cierre de la próxima sesión, situación actual en la que nos encontramos. • Información de liquidaciones: información relacionada con garantías, cobros y pagos.
RF-05	El sistema debe permitir consultar las transacciones firmadas. Debido a que el número de estas es muy elevado (ya que según el sistema que propongo, el número de transacciones sería igual a las actuales multiplicadas por seis que es el número de Px que hay en la actualidad), el sistema debe permitir realizar un filtrado por agente y fecha. Se debe conocer que por cada operación que realice cada agente, se genera una transacción firmada. Esta transacción podrá tener valor OK o por el contrario estado NO-OK dependiendo si el formato de la información que se envía en cada operación es correcto.
RF-06	Cada uno de los actores definidos en el paso correspondiente, los cuales de una manera u otra son los actores del sistema, debe poder acceder a este para interactuar con él. Para ello, cada usuario dispondrá de una clave única e intransferible que les identificará en el sistema. Además, el sistema para cada uno de los roles, deberá mostrar la información correspondiente, es decir, los usuarios con rol de operador/Px coordinador/ingeniero de pruebas visualizarán la información de todos los agentes, sin embargo los usuarios con rol de agente, el sistema solo les permitirá visualizar su información.

RF-07	El Operador / Px/ ingeniero de pruebas dispondrán de una clave única e intransferible para acceder a la plataforma de operación con el fin de poder interactuar con ella para las tareas relacionadas con la operación del mercado. Como se puede observar, los agentes tendrán acceso restringido a esta.
RF-08	Los diferentes sistemas deben permitir a los diferentes usuarios de estos, navegar por el menú principal y submenús, para consultar información de interés relativa al mercado diario, liquidaciones, datos de agentes y descargas entre otros. Informar que dependiendo del tipo de usuario con el que se acceda a los distintos sistemas, la información que se puede visualizar variará.
RF-09	La plataforma de acceso a la nube debe permitir a los operadores y por consiguiente al Px coordinador subir a la nube mediante este sistema el conjunto de fichero de capacidades recibidos de los diferentes TSO's. Además, el sistema debe permitir subirlos en nombre de cada Px, ya que de no ser así, cada fichero machacaría al anterior, además de que es de gran importancia para el casador reconocer a que se corresponde cada fichero de capacidad.
RF-10	El sistema debe permitir a un agente realizar una reclamación en el periodo de tiempo fijado, en el caso de que no esté de acuerdo con algunas de las operativas del mercado diario.
RF-11	La plataforma de operación debe poder permitir enviar un mensaje de texto al conjunto de agentes con el precio resultante de la casación.

Cabe mencionar que este último requisito funcional es otra de las funcionalidades añadidas, ya que considero de importancia que los agentes reciban el resultado de la casación, ya que de esta manera los agentes no están condicionados a estar delante de un ordenador para visualizar el precio ni tampoco disponer de una tarifa de internet para entrar en la web.

18.3.3 Requisitos no funcionales

Como ya conocemos, cuando hablamos de requisitos no funcionales, se refiere al conjunto de estos que especifican las propiedades emergentes del sistema como disponibilidad, tiempo de respuesta, rendimiento del sistema y capacidad de almacenamiento entre otras. A continuación se muestra un cuadro con un conjunto de requisitos asociados a los sistemas que se deberían utilizar:

REQUISITOS NO FUNCIONALES		
IDENTIFICADOR	TIPO	DESCRIPCION DEL REQUISITO
RNF-01	De seguridad	Los usuarios del sistema otorgados con condición de agente deben poder acceder al sistema así como a todo el conjunto de información disponible en este, siendo esta segura, transparente (no se debe ocultar ningún tipo de información) y mostrada a

		tiempo real (sin retrasos).
RNF-02	De rendimiento	Cuando el número de usuarios accediendo al sistema sea el máximo, el tiempo de respuesta de este no será superior a cinco segundos.
RNF-03	De rendimiento	Los tres sistemas de los que venimos hablando deben poder estar accesibles para todos los usuarios las 24 horas del día y los 365 días del año, entre otras cosas porque el mercado diario se lleva a cabo todos los días y la inserción de oferta para este se puede realizar a cualquier hora del día.
RNF-04	De fiabilidad	En el caso de existir un fallo de conexión, el sistema no debe perder los datos del agente, es decir, debe de recuperar datos como las unidades ofertantes dadas de alta en el sistema o las ofertas presentadas entre otros.
RNF-05	De software	De cara a los agentes, el web principal deberá funcionar y visualizarse correctamente con el navegador Microsoft Internet Explorer versión 8 (con vista de compatibilidad deshabilitada) y 9 (recomendada) de 32 bits. Así como también el sistema operativo recomendado es el Windows 7 y la máquina virtual de java Sun debe tener la versión 1.6.0 update o superior.
RNF-06	De hardware	Será necesario disponer de un lector de tarjetas, ya que cada agente dispondrá de una tarjeta inteligente y una clave, ambas cosas necesarias para el ingreso en el web principal.
RNF-07	De soporte	Debido a que los agentes deben realizar una configuración de su puesto de agentes, debe existir documentación explicativa del proceso a seguir. Además, en caso de necesidad de actualización de versiones por ejemplo de Java, estas serán notificadas a través del sistema. En caso de problemas a la hora de configurar el puesto, los agentes deben disponer de ayuda las 24 horas al día y los 365 días al año.